Understanding the Noticeability and Distraction of Interactive Highlighting Techniques

A thesis
submitted in partial fulfilment
of the requirements for the Degree
of

Doctor of Philosophy

in the
University of Canterbury

by

Joshua Leung

Supervision Committee

Senior Supervisor:	Professor Andy Cockburn
Associate Supervisor:	Professor Tim Bell

Department of Computer Science and Software Engineering
University of Canterbury

2017

Abstract

Highlighting techniques are a diverse class of visual communication techniques that make users aware of salient information in a timely manner. Any visual effect can potentially be used and manipulated to create highlighting effects given the right context, making the design space for highlighting techniques broad and rich. Although highlighting techniques are a common and important part of user interfaces, there is a lack of understanding about how to select, apply, and control their effects for achieving the best results. For example, designers need to balance some fundamental tradeoffs between ensuring that important/urgent information is able to capture the user's attention (i.e. desired *noticeability* of a stimulus), while reducing the risk that the user's attention is needlessly diverted away from their task (i.e. undesired *distraction*). However, the lack of understanding of how noticeability and distraction relate to each other, along with not knowing how we can manipulate the techniques to affect the balance between these complicates the design process.

To address this knowledge gap, this thesis provides contributions in three key areas: 1) A structured design framework for describing highlighting techniques in terms of their construction and control; 2) An empirical method and two experiment protocols for measuring both noticeability and distraction; and 3) Empirical data about the noticeability and distraction effects of highlighting techniques.

The first part of this thesis reviews current understanding of highlighting techniques, their effects, prior methods of measuring those effects, and underlying human factors. It also presents our new structured design framework – Parametric Control and Construction of Highlights (PCCH) – for describing highlighting techniques in a concise and objective way, using parameters to accurately specify highlighting technique configurations.

The second part of this thesis presents an empirical method for measuring both noticeability and distraction. This method was validated by conducting two user studies. In the first experiment, participants performed an abstract visual search task where they had to quickly drag a disk onto a cued target in the presence of 0/1/2 instances of four commonly-used highlighting techniques presented in different configurations. The second experiment was a dual-attention task where participants performed a dot-following task while detecting the appearance of highlighting techniques (in the form of Animated Window Borders). Task performance, eye tracking, and subjective experience data from these experiments are presented and analysed. Noticeability and Distraction metrics were computed from Task Performance data.

Acknowledgements

There are many people I'd like to thank for their help and support during my PhD journey. Without them, this thesis would not have been possible. As such, this thesis is a testament to the saying, "*it takes a village to raise a child*".

First, a big thank you to my supervisor, Professor Andy Cockburn, for supervising my research and opening my eyes to the world of HCI. Thanks in particular for your guidance, insights, patience, and annoyingly-exacting attention to details all these years. Thanks also to my associate supervisor, Professor Tim Bell, for your help with all the paperwork, and for helping me find ways to conquer writer's block.

To Mum and Dad: Thanks for everything! Without your love, support, encouragement, and wisdom – especially through some of the toughest stretches of this project – this thesis would not exist.

Thanks to all my friends and long-time colleagues in the Blender Community, for all your boundless positivity, energy, and inspiration. Thanks for being there when I needed a creative outlet to let off steam and recharge my depleted energy levels. Thanks also for reminding me what got me into Computer Science in the first place, and for helping me recapture my missing-in-action productivity mojo. Looking back, the seeds for this thesis were planted a few years ago as I struggled with the design for Blender's "Autokey Warning" indicator, confounded by why some designs seemed hopelessly ineffective, while others were overly distracting.

Thanks to all the COSC staff and students for a memorable few years. As one of the last pre-quake students, it's certainly been an interesting time, made even more interesting with all the remediation work that's been going on in the building.

- To *Geela, Tieta, Hayden, Sam W.* and other friends – Thanks for all the interesting discussions and fun times hanging out! It always made my day running into you guys around the department.
- To *Stephen, Joey, Sylvain, Matthieu, and Philip* – Although our time together in the HCI lab only overlapped for a few months, thanks for setting high standards (geez guys, you're a tough act to follow!), giving me a taste of the amount of dedication required to excel at the highest levels, and for inspiring me to embark on this path. Thanks to Stephen and Matthieu, who helped mentor me during my first few months when Andy was away on sabbatical jaunting around Europe. It was great to have worked in a world-class lab together with you guys.
- To *Moffat* – Thanks for showing me how to use the Tobii, and for helping liaise with Tobii to figure out those puzzling timestamp problems!
- To *Yalini* – Thanks for convincing me to start tutoring (initially for SENG301 and COSC368, and later for letting me try my hand at COSC121 too), and also for continuing to hire me again each year! Though it's sometimes been time consuming (I still shudder at the thought of marking another term's worth of 301 reports ;) it's been a really rewarding experience.

- To *Dr Jeff Johnson* and *Dr Scott Grissom* – Thanks for coming to Christchurch to teach 368 (in 2013 and 2014 respectively), giving me the invaluable opportunity to experience multiple ways of teaching an undergraduate-level "Human Computer Interaction" course. I'm glad to be one of the lucky few people who can claim to have experienced and/or been involved with – in one capacity or another – in four (!) different styles of teaching this material. I'd particularly like to thank Jeff for his helpful advice and for helping plug a few gaps in my knowledge that I'd been trying to fill for ages.
- To *Alex, Gillian, and Kerry* – For all your help, and for giving me the keys to the Tobii every time I needed to use it.
- To *Phil, Pete, Steven, and Joff* – For being helpful and keeping things running smoothly.

Finally, a big thank you to all the participants in my experiments. Without you, there would be no data to analyse and report! Thank you for taking part in my long, boring experiments.

This thesis was supported by the *University of Canterbury Doctoral Scholarship*.

Research Ethics

The studies performed as part of this thesis involved human participants. Every care was taken to ensure their privacy and comfort was maintained at all times. Participants maintained the right to withdraw their participation or data at any point.

The studies undertaken in Chapters 7 and 8 were reviewed and approved by the University of Canterbury's Human Ethics Committee (application number HEC 2015/130).

All participants signed paper consent forms before any log data, demographic information, or survey responses were collected. Copies of these forms are reproduced in the appendices.

Contents

1 Introduction

Highlighting techniques are a diverse class of visual communication techniques that make users aware of salient information in a timely manner. Any visual effect can potentially be used and manipulated to create highlighting effects given the right context, making the design space for highlighting techniques broad and rich. Common examples of highlighting effects include the 'bouncing' icons in the Mac OS X Dock or flashing title bars in Windows that indicate an application in need of attention, animated window expansion/contraction, the Mac OS X oscillating blue arrow that helps users find menu items, notification bubbles that alert users to incoming email, the 'red number badges' used by mobile apps to indicate unread messages, the enlarged 'Back' button in the Firefox and Internet Explorer UI's, and even the flashing caret in word processing software.

Although highlighting techniques are a common and important part of user interfaces, there is a lack of understanding about how to select, apply, and control their effects for achieving the best results. For example, designers need to balance some fundamental tradeoffs between ensuring that important/urgent information is able to capture the user's attention (i.e. desired *noticeability* of a stimulus), while reducing the risk that the user's attention is needlessly diverted away from their task (i.e. undesired *distraction*). However, the lack of understanding of how noticeability and distraction relate to each other, along with not knowing how we can manipulate the techniques to affect the balance between these complicates the design process. Rosenholtz et al. [140] found that there is considerable demand among UI designers for more empirical data on the effects of highlighting techniques (along with automated/semi-automated predictive tooling based on this data) to serve as an external/objective source of guidance during the design process. In particular, they found that it is often hard for designers to extrapolate from general "rules of thumb" to the complex scenarios they are faced with [140]. This raises the question: what do we need to know to resolve this knowledge gap, and how we can solve these deficiencies?

1.1 Key Opportunities

We argue that there are three key opportunities for bridging this knowledge gap about the fundamental tradeoffs between noticeability and distraction of highlighting techniques:

1. **There needs to be a structured framework for describing the control and construction of highlighting techniques.**
   Existing ways of describing highlighting techniques are ad-hoc, imprecise, and inadequate. Currently, highlighting techniques are described and referenced using varying terminology to describe the same effects (e.g. a "side-to-side movement" [81] in one paper may be the same thing as a "linear oscillation" [28] in another). Furthermore, the intensity/strength or nature of those effects is unclear, as this is often described using ambiguous and imprecise terms (e.g. a "Fast" movement in one paper may refer to an

animation cycle which repeats at 2 Hz, whereas another may deem that "Fast" refers to a frequency of 3 Hz or greater [29, 57]). Finally, there is a lack of a cohesive understanding of how all these effects fit together (i.e. most attempts [37, 106, 119] have followed the "Graphic Design textbook" approach [102, 37, 45, 106, 159] of considering visual effects such as Colour, Shape, Size, Motion, and Texture as being independent, non-interacting, non-combinable effects that are used in isolation, instead of being used as parts of a visually complex element in a UI).

2. **There needs to be an empirical method for measuring both noticeability and distraction of highlighting techniques.**
   Research so far has largely focussed on just noticeability in isolation, without considering other effects such as distraction, or the interactions between them. While some studies *have* included measures of distraction, they have been subjective measures collected separately from the main experiment [29], or were studied in isolation [121]. We argue that it is necessary to study both concurrently, by empirically measuring the effects of both noticeability and distraction within the same experiment. This makes it possible to identify cases where there is a non-proportionate relationship between noticeability and distraction (e.g. a highlighting technique that causes a small increase in noticeability and a larger increase in distraction).

3. **There needs to be more reusable empirical data about the noticeability and distraction effects of highlighting techniques, so that designers can use/refer to this for objective design guidance**.
   While many studies in the HCI and Perception literature have examined the noticeability of different visual effects, it is often not clear how these results can be applied to a given highlighting technique instantiation. This problem is accentuated by the fact that there are often fundamental differences between the experimental method/protocols used (e.g. visual search [99] versus contrast matching [161]), the units of the measurements (e.g. Lumens/Watts [161] versus RGB pixel brightness values [82, 177]), and the degree of internal versus external validity (i.e. how "abstract" or "concrete/realistic" the tasks and stimuli are). Therefore, practitioners are forced to resort to using generalised heuristics and "rules of thumb" [140].

1.2 Our Approach

This thesis lays the foundations for addressing these needs. First, we developed a design framework for describing/modelling the construction and control of highlighting techniques. Second, we developed two experiment methods for measuring the noticeability and distraction characteristics of techniques identified using the framework. Finally, we verified the effectiveness of these experiment methods by conducting user studies to empirically measure the noticeability and distraction of different highlighting techniques.

This work is important because the literature on Highlighting Techniques currently spans multiple domains including *Graphic Design* [102], *Information Visualisation* [159, 37, 110, 106, 28], *Novel UI Techniques* [173, 67, 10, 64], *Information/Cyber Security* [12], *Notifications* [119, 120, 118], and *Psychology/Perception* [39, 139, 121, 72, 142, 140, 141]. As a result, there is a considerable amount of duplicated effort and "information siloing" [157] due to each domain using a slightly different set of terminology and keywords, with little evidence of cross-pollination of relevant knowledge.

This thesis attempts to unify all of these strands of research by identifying the common body of knowledge underpinning these. On top of this foundation, we then address some of the knowledge gaps most relevant to the Human Computer Interaction (HCI) community.

1.2.1 Interactive Highlighting Techniques

In this thesis, we introduce the concept of "*Interactive Highlighting Techniques*" (IHT) as a superset and extension of "*Highlighting Techniques*" (HL). The main difference between the two is that IHT's represent the full range of temporal and interactive behaviour exhibited by highlighting techniques used in UI's. For example, see Section 4.2.3 for details about how multiple individual highlighting techniques may be combined together to form a single IHT (i.e. one HL for each of the IHT's states). This leads to the following definitions:

Definition 1
*A **Highlighting Technique** (HL) is a visual communication technique that makes users aware of salient information in a timely manner.*

Definition 2
*An **Interactive Highlighting Technique** (IHT) is a group/unit of highlighting techniques used in conjunction with each other to make users aware of salient information in a timely manner as part of a computer-based user-interface.*

1.2.2 Noticeability and Distraction

This thesis uses "Noticeability" and "Distraction" as key metrics for measuring the effectiveness of highlighting techniques for two main reasons:

1. **"Effectiveness" is a multi-dimensional quality** – It is tempting to consider the "effectiveness" of a highlighting technique to only refer to how well it can attract user attention to "an item of interest" (i.e. noticeability only). However, there is often a mismatch between what the user may be interested in, and what the system deems interesting to the user [126]. Therefore, the *unintended consequences* of a highlighting technique must also be considered when evaluating its effectiveness. Candidates for such unintended consequences include distraction, annoyance, and interruption.

2. **Usage in Prior Literature** – The combination of noticeability and distraction measures have been used in some prior studies (e.g. [29, 28]). However, these prior studies only used subjective experience measures for their analyses. To our knowledge, no prior studies have used performance-based measures of *both* noticeability and distraction to analyse and compare highlighting techniques.

In this thesis we define “Noticeability” and “Distraction” as follows:

> **Definition 3**
> ***Noticeability*** *represents the perceptual impact and saliency of a highlighted item.*

> **Definition 4**
> ***Distraction*** *refers to the undesired effects of a highlighting technique. These undesired effects may include performance degradation and annoyance.*

These definitions were chosen as they provide a practical and actionable path towards using these effects as metrics of highlighting effectiveness. Specifically, there is an explicit link between these metrics and user performance measures which could be used to objectively quantify these effects:

- *Noticeability* is linked to how quickly the user can detect/identify a highlighted item (i.e. more noticeable items should require less effort to detect, and should therefore be able to be noticed faster).
- *Distraction* is linked to the amount of *performance degradation* observed (i.e. how much worse the user performs their primary task when a highlight is present). This implies that a highlight that can be easily noticed while still allowing the user to ignore it with minimal negative effects on their primary task performance, would be deemed superior (less distracting) than one where the user has to continually attend to the highlight. For example, the user would be able to quickly determine that the highlight was non-urgent/unimportant, allowing them to solely focus on their primary task.

The links between performance measures and our definitions of noticeability and distraction make it easier for the HCI community to develop experiments for empirically measuring these effects. Subjective experience metrics (such as annoyance) can be used to supplement our understanding of the desired (noticeability) and undesired (distraction) of highlights (e.g. annoyance is a unwanted effect of highlighting, making it another dimension/component of the overall “distraction” effects a highlight causes). Part II of this thesis discusses how this can be done in more detail.

1.2.3 Key Hypotheses

In this thesis, we sought to prove the following key hypotheses about the use of Noticeability and Distraction as measures of highlighting technique effectiveness.

Noticeability and Distraction Can Be Objectively Measured

The first key objective was to show that it is possible to use performance-based measures to objectively measure noticeability and distraction as part of the same experiment:

H 1.1
The relative quality of highlighting techniques can be analysed through measurement of their emergent Noticeability and Distraction. That is, highlighting technique A is superior to technique B if

$$D_A \leq D_B \text{ when } N_A \geq N_B$$

where D is a measure of how distracting the highlighting technique is, and N is a measure of how noticeable the highlighting technique is.

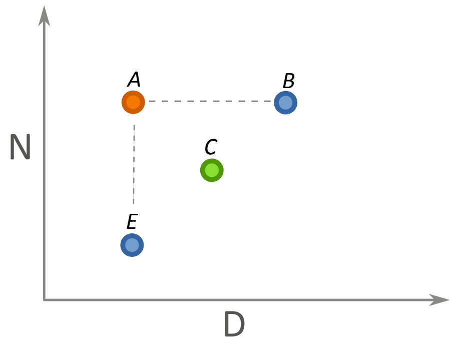


Figure 1.1: *Each dot represents a highlighting technique. A is the best technique here, as it is more noticeable and less distracting than all the others. Techniques B and E are respectively equally noticeable/distracting as A, but are otherwise worse in terms of the other aspect.*

Figure 1.1 provides a few example points that characterise how noticeability and distraction may hypothetically be used to rank the relative quality of different techniques. These hypothetical points show that:

- Technique A is the best overall, as it has the highest noticeability for the lowest distraction
- Techniques B and E are worse than A:
  - Although B is as noticeable as A, it is more distracting (and thus less desirable).
  - Although E is also not very distracting, it is less noticeable too. So, although it is not more distracting than A, it is less useful in practice, and is therefore a lower quality technique.
- Technique C is worse than A, as it is less noticeable and more distracting. It is also "worse" than B and E in some respects (i.e. it is less noticeable than B, and more distracting than E). However, in certain cases, it may still be "better" overall. For instance, C is more distracting than E, but it is also more noticeable, while C is less noticeable than B but is also less distracting.

More Noticeable But Less Distracting Techniques Exist

The second key objective was to use these metrics of Noticeability and Distraction to show that it is possible to find a pair of highlighting techniques (A and B) where one technique (A) is more noticeable but less distracting than the other (B).

> **H 1.2**
> *Across highlighting techniques, measures of noticeability do not always increase monotonically with increasing distraction. Therefore, the following condition should hold:*
>
> $$\exists H_x, H_y (N_x \geq N_y \quad and \quad D_x < D_y)$$
>
> *That is, there exist two highlighting techniques, H_x and H_y, such that H_x is more noticeable and less distracting than H_y (i.e. H_x is superior to H_y). The relationship between these techniques is shown in Figure 1.2.*

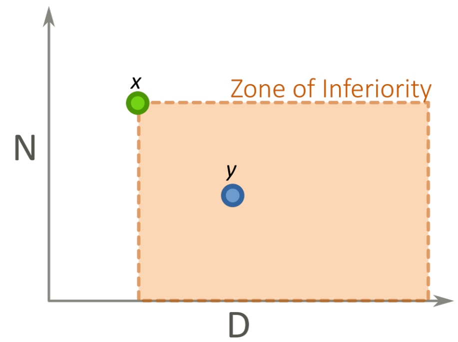


Figure 1.2: *Illustration of **H1.2**. The dots labelled x and y represent the hypothetical highlighting techniques H_x and H_y respectively. As can be seen, technique H_y exists within the "Zone of Inferiority" of H_x (i.e. it is less noticeable and more distracting).*

1.3 Key Contributions

This thesis presents the following key contributions:

1. A design framework for describing the construction and control of highlighting techniques.
2. An empirical method for measuring both the noticeability and distraction characteristics of highlighting techniques in the same experiment.
3. The results of an experiment validating that the noticeability and distraction of several commonly used highlighting techniques can empirically measured and compared.
4. The results of an experiment empirically measuring the noticeability and distraction of highlighting techniques in the form of animated window borders.

5. The finding that, as expected, Noticeability and Distraction are generally positively correlated. However, Noticeability and Distraction are not strict covariates: it is possible to find highlighting techniques which are equally (if not more) noticeable but less distracting than a given technique.

1.4 Structure of This Thesis

This thesis is divided into three parts: I) Foundations; II) Methods, Analyses, and Studies; and III) Discussion and Conclusions.

The first part (*Foundations*) presents a structured review of the prior literature and background knowledge necessary for understanding highlighting techniques. First we review the underlying human factors (Chapter 2) governing how we respond and react to highlighting techniques. Second, we review the existing literature on highlighting techniques (Chapter 3). Third, we review prior design frameworks, and present our new design framework for describing the construction and control of highlighting techniques (Chapter 4). Finally, we summarise the insights gained from these chapters (Chapter 5).

The second part (*Methods, Analyses, and Studies*) presents our methodology for measuring the noticeability and distraction highlighting techniques (Chapter 6). It also presents the results of two studies using these principles to measure the noticeability and distraction of common highlighting techniques (Chapter 7) and highlighting techniques used as animated window borders (Chapter 8).

The final part (*Discussion and Conclusions*) discusses the findings and conclusions of the work presented in this thesis (Chapter 9). We also discuss directions for future research.

Part I

Foundations

2 Human Factors for Interactive Highlighting Techniques

To understand how and why users have particular responses to highlighting techniques, it is necessary to examine the underlying human factors and phenomena governing how we perceive, process, and respond to visual stimuli.

Central to understanding the human-computer interaction issues affecting highlighting techniques is the interplay between *vision*, *perception*, and *attention*. This section summarises how the human visual processing and attention pipeline acts as a funnel that filters incoming stimuli into ordered sets of events that we respond to. Although we have a wide visual field, structural limits of the eye mean that we can only "see" a very small portion of the visual field at a time (i.e. "*foveation*") [159]. We are also only able to actively focus on a single task/thought at a time [95], meaning that there is necessarily a tradeoff between where we look/focus our attention and what gets omitted. These tradeoffs result in *Change and Inattention Blindness* [139] – a pair of well-known phenomena which impair our ability to perform *supervisory control* tasks reliably [159]. The noticeability and distraction characteristics of different highlighting techniques are therefore the result of how well those techniques balance the relevant tradeoffs to overcome Change/Inattention Blindness.

There is also the question of semantics and affordances. Humans have a keen ability to associate meaning with different types of stimuli, whether these are visual (e.g. associating glyphs and shapes with letters and logos respectively), auditory (e.g. speech and music), haptic/touch (e.g. rugged versus smooth, hot versus cold, fast versus slow buzzing), taste (e.g. associating certain combinations of flavours with a favourite dish for instance), and smell (e.g. sulphur and sewerage). However, all these associations need to be learned to be understood. As a result of individual differences in cultural background and/or other associations that have been learned over time, humans assign different semantic meanings to different stimuli, and this can have an effect on the viability of using those types of stimuli for highlighting techniques.

In this section, we first provide an overview of the core set of low-level mechanics and the interplay between them which governs how we perceive, process, and respond to highlighting stimuli (Section 2.1). This is followed by a review of studies which have identified different sets of consequences of those low-level mechanics, with a focus on the implications of these phenomena on how users interact with highlighting techniques and how we can design around these issues (Section 2.2). Finally, we provide an overview of what is known about the high-level semantic interpretations of different types of visual effects and how these interpretations can affect or be used in highlighting techniques (Section 2.3).

2.1 Overview of Low-Level Mechanics – Vision, Perception, and Attention

Given that highlighting techniques are created using visual effects, it makes sense to begin our discussion of the underlying human factors by considering how highlighting stimuli pass through the visual processing pipeline and are processed by our thought processes. That is, we address the problem of how humans perceive, process, and respond to highlighting stimuli.

There are two parts/components in this pipeline: Visual Perception and Attention.

- **Visual Perception** concerns how visual stimuli are detected by the eyes (including reasons why certain stimuli may be more easily noticed than others).
- **Attention** concerns what happens once a stimulus has been perceived; it controls how we respond to the stimulus by dictating whether we shift our focus towards the highlight (affecting how we perceive the world, as discussed below in Section 2.1.1) or whether the stimulus is ignored instead.

The *interplay* between these components governs how highlighting techniques feed into our "sense-think-act" feedback-loops [44]. That is, our field of view controls what sorts of information (or signals competing for our attention) we are able to detect. We may choose to direct attention towards signals detected by the visual system (or not), depending on our goals and/or the salience of the signal. By changing where we direct our attention, our field of view also changes, which makes some stimuli easier to detect, and others harder to detect as a result.

2.1.1 The Eye

Our ability to resolve fine details (*visual acuity*), differentiate between different colours, and to detect fine motions varies considerably across the visual field due to the non-uniform distribution of light sensitive cells in the retina [29]. The majority of cones (i.e. colour sensitive photoreceptors) are located in a small region in the center of the retina known as the *fovea*. The corresponding region in our visual field (dark region in Figure 2.1) is notable for being the region where our vision is the sharpest and is most sensitive to colour variations [159].

In contrast, the rest of the retina (i.e. "*peripheral vision*", light blue region in Figure 2.1) is primarily covered in rods (i.e. light/dark sensitive photoreceptors). There are also a smaller number of cones, but those are spaced further apart. It is well known that various measures of visual sensitivity to stimuli are substantially lower in our peripheral vision than in the fovea. For instance, prior studies have found that human observers are only able to detect one tenth of the visual detail at an eccentricity of 10 degrees (where *eccentricity* is the visual angle of a target relative to the current fixation point) [148], and that we are effectively colour blind in our peripheral vision due to the low density of cones [29]. This effect is known as *foveation*.

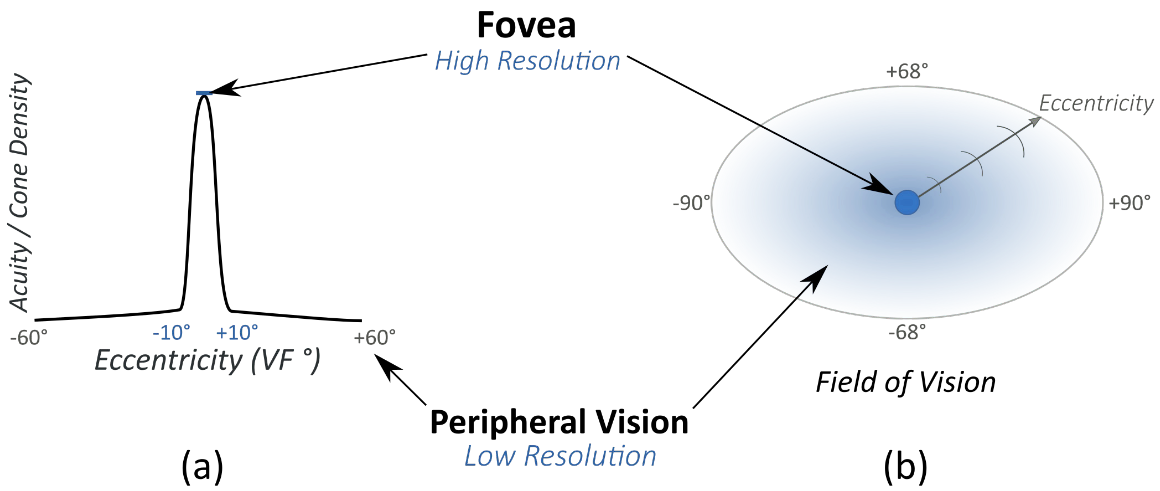


Figure 2.1: Illustration of the relationship between Foveal and Peripheral Vision (not drawn to scale). (a) Approximation of Visual Acuity (i.e. Resolution) or Cone Density as a function of Eccentricity (i.e. angular distance from center of visual field). (b) Illustration of human visual field. Intensity of shading indicates the relative sharpness/resolution of vision within each region.

Design Insight 1
Colour and fine detail can only be accurately detected in a small region in the center of our visual field corresponding to the fovea. In peripheral vision – and particularly towards the edges – our ability to detect colour differences and fine detail is poor, and degrades as the eccentricity (or distance to the focal area) increases.

There are three types of "cone" photoreceptor cells in the retina. Each of these is sensitive to a wide range of visible light wavelengths, but each is particularly sensitive to particular sets of wavelengths. One of these types of cones is most sensitive to "red" colours, another to a range of colours from red to green, and the third to "blue" colours [159]. Cones sensitive to red and/or green colours are more common than blue ones [159]. This property is often cited as one of the reasons why blue is often used for user interface "chrome" (i.e. backgrounds and frames) to avoid distracting users and to provide a "calming" effect, while orange/yellow colours are used to attract attention [19]. It should however be noted that some users suffer from various forms of colour blindness, where one or more types of cones are malformed due to genetic defects, resulting in difficulty distinguishing between colours usually detected by two different types of cone [16].

Design Insight 2
Humans are more sensitive to red/orange and green colours, and less to blue due to the differences in the number of cone cells sensitive to each of those colours. Care needs to be taken if using red/green and yellow/blue colour contrasts to indicate different categories, as colour blindness suffers have significant difficulty trying to distinguish between those pairs of colours.

Movements of the eye can be divided into three types/stages:

1. **Fixation** – This is when the eye is directed towards a particular target for a short period of time, placing the foveal region over the target so that it can be imaged in detail. [159]
2. **Saccade** – Saccades are short and fast "darting" movements of the eye from one target to another. Once initiated, the destination of a saccade cannot be changed or aborted. During a saccade, the user is effectively blind (and cannot notice any changes which occur during the saccade). However, the brain "stitches up" our perceptions of what the visual field looks like before and after the saccade, so that we do not notice that the saccade took place. [159]
3. **Smooth Pursuit** – Smooth pursuit motion is where the eye moves continuously to smoothly follow a moving target. [159]

The eye typically alternates between fixations and saccades, especially when reading. Smooth Pursuit however only applies in certain scenarios where there is a real moving target that the user is following.

According to Hoffman [85], the current understanding is that there is a lag of 100-250 ms between visual attention being drawn to a target and when the eyes focus on that target. This supports the notion that the eye is constantly repositioned to fixate on a region of interest, placing the foveal region over the region of interest so that the details can be imaged/inspected in detail [159]. In the Handbook of Perception and Human Performance [39] (page 42-39, section 5.3.2), this is referred to as the "eye point of regard".

Design Insight 3
The eyes move to position the foveal region (i.e. the most region most sensitive to colour and detail) over the target so that it can be examined in detail. It also changes where everything else lies in peripheral vision, and by extension, how well other targets can be detected at different times.

To better understand the relationship between eye movements and visual attention, it is helpful to use the Ware's spotlight metaphor for explaining the relationship between the orientation of the eye/fovea and where a user's visual attention is directed [159]. In this model, the beam of the spotlight represents the limited field of view of the foveal region, and the area illuminated by the beam is what can be observed by the user at that point in time. Thus, to see the rest of the scene, a user must therefore decide where to shift their attention, and that is driven by clues detected in peripheral vision outside the illuminated beam.

2.1.2 Visual Search

Visual Search refers to perception tasks where the user is trying to find a particular item from a display [159]. It often occurs in interfaces which the user is unfamiliar with and/or interfaces where the user is unable to anticipate where the item they want is located [144]. That is, users are unable to rely on spatial memory to enable fast revisitation of familiar targets and landmarks [144, 145]. Therefore, designers seek to reduce the need for users to perform visual search, as it is known to be a bottleneck for user performance [67].

Highlighting techniques can be used to reduce (or avoid) the need for users to perform visual search. They do this by creating "*pop-out effects*". Pop-out effects make highlighted items easier to identify by increasing their visual salience and/or by taking advantage of how humans perceive and identify salient items during the visual search process.

Treisman and Gelade's "feature integration theory of attention" [154] is a useful starting point to understanding what types of visual stimuli may be useful for acting as salient targets for the eye. Their thesis is that there are two types of visual search processes: Feature Search, and Conjunctive Search (see Figure 2.2).

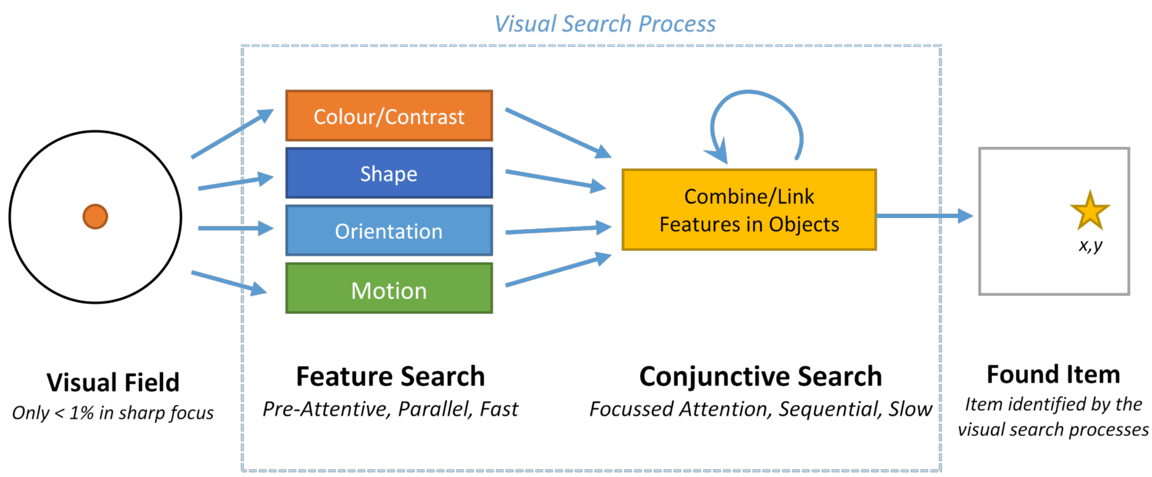


Figure 2.2: *Overview of how visual search works as per Treisman and Gelade's "Feature Integration Theory". Visual stimuli detected by our eyes are processed in two stages (i.e. Feature Search and Conjunctive Search), where salient regions are detected and these are analysed to determine whether it matches the target. The end result is that the location of the target item in the visual field is detected. Visual attention may then be directed towards this item if we decide to pay attention to it.*

Feature Search is a pre-attentive process. It is performed in parallel by a set of "feature detectors" which detect differences between objects/visual stimuli in terms of colour/contrast, shape, orientation, motion, and other simple graphical characteristics. Each detector is only able to identify one type of stimulus, but can perform such detections really quickly. Together, these characteristics of feature search mean that this type/phase of visual search can be performed really quickly.

Conjunctive Visual Search is much slower as it requires users to consciously consider and inspect the differences between candidate targets identified during the initial Feature Search process. This visual processing pathway is activated when multiple attributes are simultaneously encoded using two or more types of visual features, and the user needs to identify a specific combination of features to complete their task. An example of where this happens is if colour (e.g. red and green) and shape (e.g. square and circle) are used to encode two different properties at the same time. The highlighting technique in this case would require the user to identify the item where a certain combination of these two features are applied to the same object (e.g. red square, or green circle). This type of visual search can be observed when users are asked to retrieve a specific item from an unfamiliar list or grid based display.

The main conclusion that designers should draw from Treisman and Gelade's model [154], is that when highlighting techniques need to quickly attract the user's attention, they should try to prevent the user from needing to perform Conjunctive Visual Search. This can be done by creating pop-out effects, meaning that highlighted items can be identified during Feature Search (fast) without the need for any Conjunctive Visual Search (slow) to be performed.

Design Insight 4
Highlighting techniques that need to be quickly and easily noticeable should create pop out effects to benefit from the fast Feature Search phase of perception and avoid slow Conjunctive Visual Search. To do this, designers should focus on using ***unique features*** *instead of* ***unique combinations of features****. For example, a quickly noticeable highlighting technique could be to make the highlighted item be "the only red item in the scene". In contrast, an ineffective technique would be to make the highlighted item be "the round item with a green border" (in a field containing with a mixture of green/blue items, and items with rounded/sharp edges).*

This two-stage model is very similar to the neurobiological model used by Itti et al. [90] which models how the human visual system would process a scene to identify salient regions in the visual field. In Itti et al.'s model, the Feature Search process corresponds to the feature maps which are computed by a series of different specialised feature detectors (mirroring the "V1" – visual cortex – cells [141, 159]) running in parallel; the intensity value of pixel in such maps determines how "salient" the detector deemed the corresponding pixel/area in the visual field. The Conjunctive Search step then corresponds to the aggregation and filtering steps which determine the areas of interest across the feature maps. This ultimately culminates in the "winner takes all" selection of the candidate that is deemed to be the most salient. There are significant similarities between these two models. Itti et al.'s model has been shown to be quite accurate at predicting the most salient targets in a scene at any point in time (when comparing the model's predictions with eye gaze data) [90, 89].

2.1.3 Attention – A narrow beam of focus

Attention is a precious and limited resource [159]. It can be thought of as representing what the user chooses to direct their limited cognitive resources on while ignoring all other stimuli [159, 95]. These cognitive processing resources include our active/conscious thought process and reasoning abilities, short term memory, as well as where our eyes are focussed (as described in Section 2.1.1). There are several links between attention and how users respond to highlights: noticeability relates to how well the highlighting technique can gain the user's attention, while distraction relates to whether the highlight affects their concentration (i.e. affecting how well they can pay attention to something of greater interest/relevance).

The current understanding is that **attention can only be directed towards a single target at a time** [159, 95]. Kahneman's *Capacity Theory* [95] argues that even though we may be able to perceive multiple stimuli concurrently, we can only actually focus on or process a single one at a time. That is, when multiple stimuli arrive at the same time, they get scheduled to be attended to in quick succession, with some being given priority over others (e.g. visual signals may take priority over auditory) [95]. Capacity Theory argues that **attention focussed on one target is lost or retargeted from elsewhere** [95]. Therefore, our apparent ability to "multitask" is only an illusion. When we "multitask", we are actually quickly switching between several different contexts/tasks, instead of concurrently focussing on multiple tasks [159].

The "*Executive Control System*" controls what we focus on or think about, and has long been linked to the *Prefrontal Cortex* [101]. It has been found that the *Prefrontal Lobe* can only process a single thing at a time, but is able to switch between two tasks rapidly [101]. There is however of cost of impaired mental performance for each task/context switch [95], so "multitasking" is problematic as it can impair our overall productivity if performed too frequently [114].

2.1.4 Top-Down and Bottom-Up Models of Attention

Models of human attention can be divided into two classes [95, 40]:

- **Top-Down or Task Focussed** – The Top-Down model assumes that users have a set of goals that they are trying to achieve, and that all decisions the user makes are considered in terms of whether that action would further their current goals. Top-Down models work by making estimates of what the user's goals are, and then predicting what action the user is most likely to next perform to further those goals. However, such models are rare in the current literature, as the problem of anticipating user goals is ill-defined, making it difficult to begin creating such models [40].
- **Bottom-Up or Saliency Driven** – Bottom-Up models work up from the stimuli instead. They work by identifying the most salient stimuli as being the ones that the user is most likely to focus on next. These approaches are backed by decades of psychology research, which have resulted in the development of predictive/simulation models such as that of Itti et al. [90] and Rosenholtz et al. [142].

Human attention is likely to involve a combination of these models working in concert with each other. When highlights are detected by our senses (i.e. Bottom-Up processes), they are added to the queue of items/tasks that the Top-Down/Executive System (i.e. conscious thought processes) need to consider next. The Executive Control System (i.e. Prefrontal Lobe [101]) can then decide whether to redirect attention to this stimulus, or ignore it. However some stimuli may be so noticeable/distracting that the Executive System cannot ignore them [176].

Yantis and Jonides [176] re-appropriated the term "*automaticity*" to refer to whether a stimulus was "strong" enough that it would override the Top-Down system's attempts to ignore or prevent attention being redirected towards that stimulus. Specifically, they claimed that for a stimulus to be considered compliant with (what we shall herein refer to as) "*perceptual-automaticity*", the stimulus needs to meet two criteria:

1. **Load Insensitivity Criterion** – The automatic process is not hindered/impaired when there are concurrent perceptual or cognitive loads placed. This is can be understood by considering the following crude metaphor based on task scheduling performance in a computer – 'automatic' processes are analogous to tasks which can be scheduled to run on some dedicated hardware; they can keep running even when the CPU (i.e.

our "conscious" thought process, which handles more generic operations) is fully subscribed to a particular task or tasks. Thus, this implies that in high-workload/high-stress environments, highlighting techniques satisfying this criterion should be able to "cut-through" all the clutter and noise simultaneously competing for the users attention.

2. **Intentionality Criterion** – The user cannot decide/force themselves to prevent the process from taking place (e.g. they cannot decide to ignore the stimulus).

Design Insight 5
Highlighting techniques which satisfy the Perceptual-Automaticity criteria (i.e. the highlighting effect is so strong that that brain effectively cannot ignore it) are useful in high-workload/high-stress environments and/or in situations where it is critical that the user does not miss/ignore the information being highlighted.

2.1.5 Stimulus Detection Outcomes – Insights from Signal Detection Theory

Our sensory systems are constantly exposed to a barrage of sensory inputs, including visual stimuli, sounds, touch/haptic/temperature, smell, and taste. Therefore we filter out inputs of lesser importance so that we can focus on what is truly important (i.e. to separate the "signal from the noise") [159]. *Signal Detection Theory* (SDT) provides many useful insights into the interactions between an observer and the highlighting techniques which may or may not be present in an interface. In particular, we focus on the most relevant concepts here: the four possible outcomes from each highlight detection decision made (i.e. True Positive, False Positive, False Negative, and True Negative), and the mental models used that govern the way we approach these interactions (i.e. Payoff Matrices) [39].

A highlighted object in a UI can be considered to be a *signal* within SDT. The user may detect or fail to detect the signal. Furthermore, the user may falsely identify something as highlighted when no highlight was intended. The possible conditions of stimulus and detection are summarised in Figure 2.3.

Ideally, users would only ever make correct detection decisions. That is, they would only detect a signal when one exists (*True Positive* or *Correctly Noticed* – Figure 2.3a), and would not detect a signal if none were present (*True Negative* or *Correctly Ignored* – Figure 2.3d).

However, various human factors mean that this is not necessarily the case. These incorrect detection decisions are the source of the "False Alarm" and "Missed" outcomes. The "False Alarm" outcome (Figure 2.3b) is generally harmless, and serves mostly to annoy and/or act as a distraction to users (e.g. the "phantom buzzing" phenomenon, where it seems to a user that their phone is ringing or has a new message/notification, only to discover that nothing had happened when they actually check). The "Missed" outcome (Figure 2.3c) in contrast may carry some undesirable penalties (e.g. death, serious injury, or loss of job/career/finances) as the user was not able to respond in an appropriate way because they did not notice the relevant warnings or status indicators.

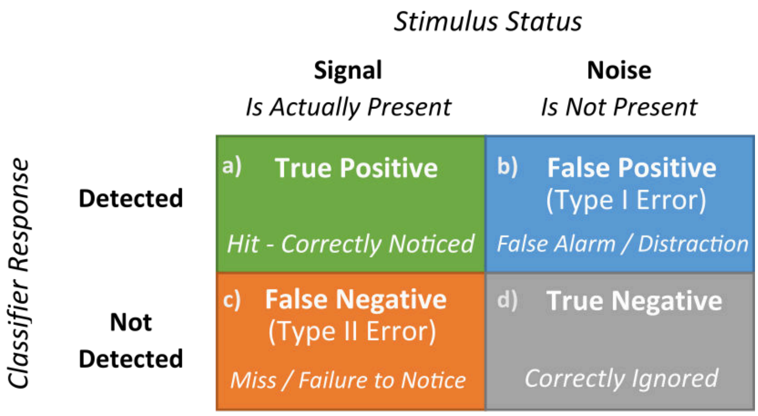


Figure 2.3: *Confusion Matrix showing the outcomes of highlight stimulus detection decisions. The first column is for cases where the highlighting stimulus is present, while the second column is for cases where the stimulus is not actually present. The first row is for cases where the classifier/user reports that it detected the presence of a stimulus, while the second row is for cases where a stimulus was not detected.*

There is also the problem of whether a highlight is highlighting something that the user is interested in. The "Correctly Noticed" label in Figure 2.3a) refers only to the fact that the user detected a highlighting stimulus that was indeed present. It does not imply that the stimulus noticed was something the user would *want* to pay attention to (e.g. advertising banners [32]). For example, consider an interface where two highlighting stimuli are present: one is used to indicate some task-relevant information (e.g. the total price of an order) and the other is an banner advert. If the user detects either of these highlights, that detection event would be classified as being "Correctly Noticed". However, in the case of the advert, since the user noticed something that holds little apparent value to them, the user treats the offending stimulus as a "False Alarm" instead.

2.1.5.1 Implications of Detection Outcomes – Payoff Matrices

Complicating matters is that user maintain a mental model of the perceived utility (or cost-benefit tradeoffs) for attending to or ignoring a stimulus. For example, one of the reasons why many users quickly learned to automatically click-through popup boxes was because the information they provided was often perceived to be of little practical value [93, 103], whereas the battery status indicator (e.g. a low battery could result in imminent data-loss at an inconvenient time) or the status indicator on a smartphone (e.g. a potentially urgent message from people they know) are more likely to capture the user's immediate attention due to the costs of ignoring them. These tradeoffs can be summarised using *Payoff Matrices*, where each Highlight Detection Outcome is assigned a score for the net benefit/cost of attending to a potential highlight [39].

Figure 2.4 shows several examples of different Payoff Matrices that may apply when different detection strategies (and the associated thresholds) are used. In each case, there is nothing to be gained or lost if nothing happens, so the "Correct Rejection" term (bottom-right corner) is assigned a weight of 0 across all the matrices.

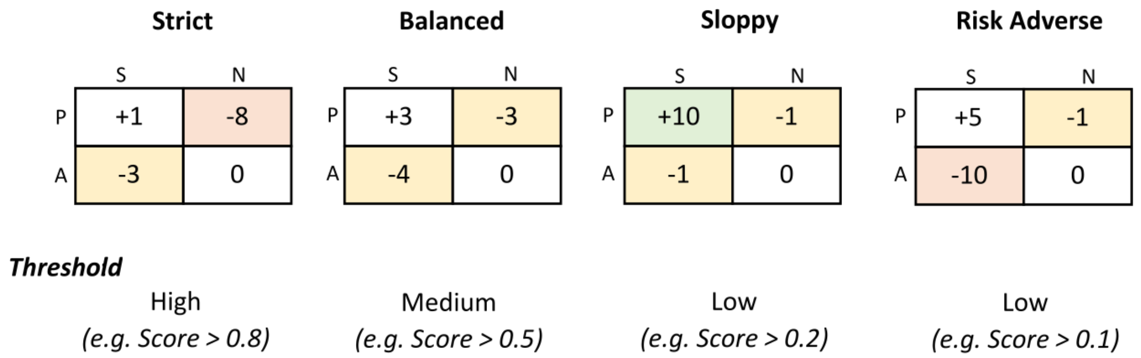


Figure 2.4: *Three examples of Payoff Matrices which correspond to different detection strategies and thresholds. These are - a) Strict, with a high threshold for responding, b) Balanced, with a medium threshold, c) Sloppy, with a low threshold, d) Risk Adverse, with a very low threshold.*

The presence of these mental models may have some effect when conducting studies of highlighting effectiveness. Care needs to be taken when considering the relative proportion of trials where highlights are used to help the user (e.g. for trials where the target item is highlighted to make it easier to identify) versus trials where highlights are used to act as decoys/distractors (e.g. for trials where we are trying to measure the distraction effects of the highlights). For instance, if participants quickly learn that the highlights are less likely to be beneficial, they may start ignoring them, thus decreasing the accuracy of any measurements of how noticeable the techniques actually are, as there is now an additional "uncertainty cost".

2.1.5.2 Detection Accuracy Metrics – True Positive Rate and False Positive Rate

True Positive Rate (TPR) and False Positive Rate (FPR) are a pair of measures which can be used to describe how accurately a highlighting technique can be detected [39]. They are typically used for other signal detection problems to assess how well a classifier (i.e. the user) can detect the signals (i.e. highlights) they are presented with. These measures can be defined as follows:

$$\text{True Positive Rate} = \frac{\text{Hits}}{\text{Hits} + \text{Misses}} = \frac{\text{Hits}}{\text{Total Signals}} \tag{2.1}$$

$$\text{False Positive Rate} = \frac{\text{False Alarms}}{\text{False Alarms} + \text{True Negatives}} = \frac{\text{False Alarms}}{\text{Total Noise}} \tag{2.2}$$

That is, the TPR (or "sensitivity") represents the proportion of cases where the user correctly detects the presence of a highlight, given that there are highlight present. Likewise, the FPR (or "specificity") represents the proportion of cases where the user incorrectly "detects" the presence of a highlight, even though no highlight was present. Together, these two ratios [1] could potentially be used to represent the effectiveness of a highlighting technique.

[1] In addition to the TPR and FPR ratios, there are also the False Negative (FNR) and True Negative (TNR) rates as well. However, it is not necessary to use these, as they are simply the inverses of the TPR and FPR (i.e. $FNR = 1 - TPR$, $TNR = 1 - FPR$)

2.1.6 Supervisory Control

According to Ware [159], *supervisory control* relationships are a type of human-computer interaction relationship where humans play a dual monitoring and control role over a complex semi-autonomous system. That is, people operating such systems would need to constantly monitor the system state for abnormal behaviour. Operators need to maintain *situational awareness* (i.e. knowledge of the current state of all the various parameters in the system) so that they are able to make appropriate decisions if a "human-interrupt signal" [159] is emitted by the system (in the form of a highlighting technique). Such relationships occur in safety-critical control systems such as those found in aircraft cockpits and power plants (e.g. nuclear power plants and also power grid control facilities), where operators need to undergo special training on how to make effective use of the information available [159].

With the rise in automation and complexity of modern computing systems, supervisory control relationships can now be found in a wide variety of consumer-grade applications and systems for use by lay-people [159]. Examples of interfaces where supervisory control relationships occur include email inboxes, messaging apps, and even some productivity suites. These interfaces make use of various highlighting techniques as indicators of status and also for notifying the user that there is information to attend to. Thus, supervisory control relationships are one of the primary scenarios that highlighting techniques are deployed in.

However, humans are not well adapted to performing continuous monitoring tasks. In particular, there are a number of well-known phenomena such as Change Blindness and Inattention Blindness [57] which mean that we can often fail to notice seemingly obvious changes. These phenomena are discussed in more detail in Sections 2.2.

2.1.7 Scanning Patterns

Users develop routinised patterns of eye movements (or "*scanning patterns*") for regularly "polling" whether different status indicators have changed, to counteract the undesired effects of Change/Inattention Blindness and other phenomena which affect our ability to perform monitoring and supervisory control roles adequately. Another reason for regularly polling different status indicators is to identify patterns of changes which may suggest certain trends which cannot be indicated by a single indicator[159]. Pilots in particular are taught to use scanning patterns (e.g. they use a "T"-shaped scan pattern for checking the set of key instruments that are arranged close to each other in a T-shaped configuration). Drivers are also taught to use scanning patterns as well in the form of sequences of mirror-checks, shoulder-checks, head-checks, speedometer checks, and regular "scanning the road ahead" checks for hazards. Operators of other systems requiring complex supervisory control relationships (such as power plants and medical scanning equipment) also receive special training on how to maintain situational awareness using scanning patterns.

Psychologists have developed models of *visual monitoring strategies* to understand and explain why users adopt particular scanning patterns [159]. According to Wickens [163], these models describe how have three components work together:

- **Channels** – A channel is a way that the system presents information to the user. Examples of channels include the use of different status indicators (e.g. a status indication widgets which may get highlighted in some way), as well as audio [38] or haptic [62] stimuli.

- **Events** – An event refers to a change/signal occurred in the channel. This often means that the state of a highlighting technique changed – from an inactive/off state to a highlighted state. An example of this is an indicator light becoming illuminated.
- **Expected Cost** – This is the cost/penalty of *missing* an event. See Section 2.1.5.1 (Payoff Matrices) for related concepts.

In general, users try to sample channels in such a way that the maximise their chances of noticing an event when it occurs, while trying to avoid the penalties of missing those events (i.e. expected cost(s)). That is, Scanning Patterns are informed by each user's *mental model* [126] of *how often things change* [159], and *how useful the information is* to the user when they encounter it [131]. To do this, the user develops knowledge about the probability that sampling a channel will yield information (i.e. event or change in some indicator) that is relevant to their tasks. We refer to the collection of these utility ratings/probabilities as the *Sampling Payoff Matrix* (SPM). The SPM is not to be confused with the Payoff Matrices introduced in Section 2.1.5.1, as the cells in SPM's represent the probability that sampling the channel will yield useful information, whereas the cells in Payoff Matrices represent the net impact (cost/benefit) gained as a result of each detection outcome occurring.

One of the implications of SPMs is that users are likely to sample items with higher scores more frequently than items with has a lower scores. The strategy used here relates to the concept of "*Information Scent*" (InfoScent) from Information Foraging Theory [131]. InfoScent is a measure of the "*perceived utility gained relative to the amount of effort required*" [131]. The implication is that an item which provides useful information more often (i.e. with a higher probability) whenever it is sampled provides better value to the user than an item which rarely provides useful information when sampled. That implies that the user chooses to attend to or check on items item which provide more value more often than those which provide less value. An example of a class of "bad" stimuli which provide low value to users are banner ads, as those techniques have been found to be frequently ignored by users [43, 32].

Mismatches between designer's model of what important to users, and the user's model of what they need/want is source of conflict which can lead to highlighting techniques being less effective than they could otherwise be. Designers can use Sampling Payoff Matrices as a tool for considering the relative utility that users would expect and/or are able to get from each piece of information they are exposed to at any point in time. Specifically, information which is found to be likely to be more important, urgent, or needed more often should be candidates for stronger techniques which will be harder to miss or which will require less frequent polling as they can successfully capture the user's attention otherwise. Gluck et al.'s [74] work on matching the strength of highlighting techniques with the utility of the information they provide suggests that there are some benefits (e.g. how "annoying" the techniques were perceived to be) to taking care of the proportionality between utility and strength.

Design Insight 6
Consider the proportionality between highlighting utility and highlighting strength. Sampling Payoff Matrices are a useful tool for considering the relative utility of all the different pieces of information that the user may be exposed to in the interface at any time.

The utilisation of scanning-pattern behaviours could be viewed as an indication that the highlighting techniques used are ineffective, as it indicates that the user does not trust that they will be alerted to critical information. For example, polling behaviours such as, “Did I remember to enable that privacy checkbox before I hit send?”, are likely indicators of potential design flaws. However, sometimes it may be necessary to utilise scanning patterns to adequately assimilate information from several disparate displays, when the information needed at any instant varies with the task being performed. For example, depending on the stage of flight, certain instruments are likely to be of greater relevance than others. In this case, the use of scanning patterns is justified as it is not possible to keep all necessary information within the focal region due to the complexity of the task domain, and the risks posed by having a unpredictable interface layout.

2.2 Human Sensitivity to Stimuli

The core mechanics described in Section 2.1 only explain why the *potential* exists for some highlights to get detected more easily than others. Specifically, we discussed how items which fall further away from the focal point are less able to be detected as well as those in the foveal region, how Feature Search allows items with unique features to be identified quickly (without the user needing to resort to slower Conjunctive Visual Search), how our attention can only be directed towards a single target at a time, and how our executive control faculties govern where we direct that attention (at the expense of sometimes ignoring stimuli which appear). In this section we build upon this foundation, and discuss in more depth the subtleties of these considerations to understand why some highlights are easier to notice (and correctly identify), while others are frequently missed.

Research about how the human visual system detects or perceives different stimuli (i.e. visual perception) is the focus of a broad branch of psychology. A considerable body of literature has been published by psychologists documenting studies examining a wide range of variables (including colour/contrast, shape, size, pattern, and background intensity), the interactions between these variables, and how these variables and interactions between variables affect our ability to perceive different visual stimuli.

There has also been a lot of research about the nature of attention in psychology. According to Kahneman [95], research on attention is divided into several subcultures focussing on areas such as “visual perception”, “speeded performance”, “physiological arousal”, and “studies of audition”. Many different models of how attention works (both top-down and bottom-up) have also been proposed.

It is clear that these topics are highly relevant to the HCI community, particularly when we are working with highlighting techniques. However, psychology research in general remains relatively inaccessible to many HCI practitioners. Common problems faced by HCI practitioners seeking insights from the psychology literature include the use of tools, techniques, and measures which cannot be easily adapted or translated for use in practical HCI applications. For instance, it is common for perception research to be conducted using highly specialised forms of abstract stimuli (e.g. one or two Gabor patches set against plain backgrounds [161]) which are often far removed from the world of complex widgets and information-rich screens that are common in UI's. Another example is how colour/brightness are measured in units of "trolands" (i.e. the amount of retinal illumination, which depends on pupil dilation and the incoming light intensity) in some studies [161], a unit which is tricky to compare with RGB and LCD backlight intensity values. Therefore, there is an opportunity here to bridge the gap between psychology and HCI research, by identifying some key insights from psychology research which are of use for HCI applications.

In this section, we present and discuss the results of studies studying phenomena which affect how users interact with highlighting techniques. These are presented in decreasing order of impact, from highest impact to lowest impact.

2.2.1 Change Blindness and Inattention Blindness

Humans are not well adapted for performing system monitoring tasks. Change Blindness (CB) and Inattention Blindness (IB) are a pair of well-known phenomena where users are unable to notice seemingly obvious and substantial changes within their field of vision [139, 113, 30, 57]. The impact of these phenomena is potentially severe, as there are cases where users will not only miss the change/highlight when it first appears, but may completely fail to register that the change/highlight has occurred at all [139].

2.2.1.1 Change Blindness

Change Blindness (CB) refers to cases where the user fails to detect changes due to "visual disruptions" [139, 30, 57]. Examples of the different types of disruptions which can cause users to fail to notice changes include:

- **Observer Movements** – The observer blinked, turned their head, or performed a saccade from one target to another.
- **Target Movement** – The plane that the highlights appear on rotated or tilted at the same time that the change occurs.
- **Obstructions** – Various physical obstructions may partially block the user's view of the scene including raindrops, mud splotches, bars/grills/nets/meshes, noise patterns, or insects (either crawling on the screen or flying in front of the user)

All of these types of visual disruptions commonly occur in real life. User movements may coincide with a highlight appearing causing the user to fail to notice the highlight. Physical obstructions of the visual field are also common (e.g. a windscreen/visor/glasses covered in rain droplets or dust/pollen). CB has been very well studied in the past decade, as it has serious implications for driver safety [57]. The insights from those studies are useful for the HCI community when working with highlighting techniques.

CB is most likely to occur when the change occurs on an item that the user is not focussing on [139]. A popular theory is that the brain replaces each non-focussed object in the visual field with a "proxy" (i.e. a simplified mental representation). Thus, if the user does not deliberately attend to the real object/region again, or if the change occurs in a region obscured by disruption, the user is not able to update their mental map of what is present in that area [57]. Therefore, the user has no situational awareness of the changed item having changed, as the old mental proxy is still in use. This problem is known as Rensink's "One Shot Paradigm" [139].

Design Insight 7
To guard against the risk of Change Blindness occurring when the user blinked or had their vision obstructed when the highlight initially appears, it may be necessary to periodically "remind" the user that the highlight is there afterwards, so that they have additional chances to notice the highlight. This is particularly necessary if the highlight is being used to communicate urgent or important information.

2.2.1.2 Inattention Blindness

Inattention Blindness (IB) refers to change blindness which occurs when the user fails to detect highlights or items *in the absence of any visual disruptions* [113, 57]. That is, there are no physical reasons why the user cannot notice the change; instead, the failure to notice highlights is IB may happen if the user is engaged in a routinised set of actions [133] or is engaged in a high-workload task [95]. The result of IB is that that the user is unable to notice any additional stimuli when they appear, even if those stimuli are extremely unusual [159, 139] or would usually create the "pop out effects" usually required to gain a user's attention [57]. Several well-known experiments have documenting this effect, such as the "invisible gorilla suit" or "invisible plane parked across a runway" [139].

Users performing a set of routinised actions may fail to notice changes due to the brain's tendencies to "adapt to" or "normalise away" repetitive stimuli [159, 163, 133]. An example of these adaptive tendencies is the way our brains start blocking out and compensating for "background noise" such as constant fan/engine sounds. An example more relevant for HCI is the way that "UI chrome" such as toolbars, statusbars, window borders, and other non-changing elements (which do not have any immediate relevance to our primary task) appear to fade away from our awareness.

A particularly problematic consequence of these adaptive tendencies is that we "chunk together" the sequence of actions and stimuli which occur over the course of performing some repetitive task [133]. That is, instead of thinking about each piece of information we are presented with, or thinking about each action that we need to perform, we tend to use high level of abstractions of these sequences of actions (e.g. to send this file to a coworker, I need to "click button 1, click-through the dialog that shows up, then click buttons 2, 3, 4"). This is why users often automatically "click-through" confirmation dialogs such as the one presented when deleting a file [103, 133].

> **Design Insight 8**
> *Highlighting techniques used to draw user attention to urgent or important information may need to be varied so that users are less likely to get used to ignoring that specific technique wherever it appears. This is necessary for countering the problem of Inattention Blindness.*

2.2.1.3 Effects of Change and Inattention Blindness

Davies and Beharee [57] studied the effects of CB and IB on mobile devices. In their first experiment (for studying the effects of CB), they found that as the number of items (e.g. icons in a menu) increased, participants were less able to detect a change if a visual disruption (flicker) occurred [57].

> **Design Insight 9**
> *Change Blindness is more disruptive in more cluttered displays (i.e. there are a higher number of items visible), as it is harder for the user to notice a change if their visual field is disrupted when the change occurs.*

In their second experiment (for studying the effects of IB), there was a 34.5% False Negative rate (i.e. this proportion of notifications were "completely unnoticed"), with True Positives detected in 3877ms on average [57]. That is, more than a third of all notifications displayed were missed by participants.

There was a significant difference between "*insertion of objects*" (i.e. a new icon appeared for 3 seconds) versus "*gradual changes*" (i.e. the text of a label was changed) [57]. Specifically, notifications using the "insertion" technique were noticed in 2034 ms, which was half the time (5317 ms) that it took participants to notice the "gradual change" one [57]. Davies and Beharee claim that this is because the insertion technqiue had a more salient "*change transient*" [57]. They define change transients as being the "apparent *pop*" that pop-out effects create.

However, there was no significant difference identified for placing notifications at the top or bottom of the screen [57], which is to be expected since both edges should be a similar distance to different parts of the game board used as the primary task. However, participants rated notifications appearing at the top more distracting [57].

> **Design Insight 10**
> *Highlighting techniques which add additional elements to the screen are more effective (i.e. they are more noticeable due to faster reaction times) at combating inattention blindness than techniques where the graphic content of an existing element is mutated.*

2.2.2 Clutter, Crowding, and Peripheral Vision

Most highlighting techniques are initially noticed (or not noticed) in peripheral vision, given that it covers the majority of the visual field. Peripheral vision is generally "worse" than foveal vision, with reduced sharpness, sensitivity to colour differences, and detection of changes in general [159]. However, there are some notable exceptions to this, such as how our ability to detect motion remains relatively constant across the visual field [159, 29]. Many studies in the literature have studied the ability of humans to detect stimuli in peripheral vision [79, 28, 86]. In this section, we focus on the issues of crowding and clutter.

Clutter is defined as the presence of other visual stimuli/objects which are also competing for the user's attention [142], while *Crowding* is the "inability to perceive objects in clutter" [162].

Crowding is most likely to occur when items are closely spaced together in peripheral vision, meaning that the viewer may be unable to notice any additional items which may show up in the same region(s) as well [149]. An example of this effect is when a driver is not able to notice a kid standing between two large road signs (placed closed to each other) while fixating on a point in the distance (see Figure 1 in [162]). Another example may be the user failing to notice a "red dot number badge" being added to an icon on a toolbar.

Bouma's Law [41] defines the critical distance for the onset of crowding [149] – that is, the minimum amount of separation between the target item and "flankers" (or distractors) around it where the viewer can successfully tell these items apart in peripheral vision [141]. It can be defined as:

$$d = bE + w \tag{2.3}$$

where d is the *center-to-center critical distance* (i.e. the minimum distance between the centerpoints of a pair of items so that an observer can discriminate between the items), E is the *eccentricity* (or how far away the target is from focal point, in visual-angle degrees), w is the "width" of target (in visual-angle degrees), and b is the "Bouma's Constant" (i.e. the slope of the curve defining the relationship between eccentricity and critical distance).

Bouma's constant, b, defines the slope of the curve for the minimum distance that items in peripheral vision need to have between them in to be able to be easily detected. Bouma originally estimated b to be 0.5. Recent studies by Strasburger et al. [149] found that b had values of 0.7 or 0.625 in contrast sensitivity studies.

Design Insight 11
Items spaced closer than a critical threshold will not be able to be detected by users in peripheral vision. This distance threshold depends on the eccentricity (or how far away the target item is from where the user is currently looking/focussing). This relationship can be approximated as $d = 0.7E$, where d is the minimum distance, and E is the eccentricity (in visual-degrees).

The exact mechanics of peripheral vision and crowding are still areas of active research. According to Rosenholtz et al., there are four competing models how visual clutter affects the way that we perceive items in peripheral vision [140]. These include Rosenholtz et al.'s "summary statistics" model for visual search [141], and Berg et al.'s model [156].

2.2.3 Interruptions – Importance versus Urgency

This section examines what is known about the impact of interruptions on the user, and how forceful interruptions should be to gain the user's attention while minimising their unwanted effects.

Several studies have found that the best times to interrupt users are at "task boundaries" or "coarse breakpoints" in a sequence of tasks/actions [27, 7, 54]. Bailey et al. found that when interruptions occurred during a task (instead of at task boundaries), task completion times were 3-27% longer than at task boundaries, twice as many errors were made, and rates of frustration/annoyance were 31-106% higher [27]. Similar effects were also observed by Adamczyk et al. [7] who found that interruptions occurring at "coarse boundaries" in tasks resulted in less frustration and annoyance. Cutrell et al. [54] found that interruptions occurring earlier in a search task resulted in participants being more likely to forget what their primary task was.

However, in practice, it is difficult to implement these ideas into functional systems. In order to determine whether a task boundary has been reached, it is necessary to have an accurate model of what the user's tasks are (i.e. a "top-down" model – Section 2.1.4). As discussed earlier, such models are difficult to build as the tasks that the user has change often/rapidly without much warning, so predicting the user's current task is a difficult problem. There have been a few attempts at developing such predictive tools [7, 69, 88].

Design Insight 12
It is better to present interruptions when users are between tasks, or towards the end of an existing task. Interruptions should however be avoided when users have just started a new task.

Interruptions may not be an entirely bad thing. Mark et al. [114] found that although users who were interrupted had higher levels of stress, they also had a tendency for faster task performance as they may have attempted to compensate for time lost due to the interruptions. Dabbish et al. studied the causes of "self-interruptions" – that is, interruptions which come about from the user remembering that they need to do something else or attend to some channel, as opposed to interruptions being caused by some external events. Dabbish et al. suggest that self-interruption may be the product of "prospective memory events" (e.g. suddenly remembering something you need to do) [56]. It could also be argued that self-interruptions are also a product of our need to "oversample" infrequently updating channels (e.g. repeatedly checking on an indicator once per minute instead of once every twenty when it actually updates, to avoid falling victim to inattention blindness) [159].

The other thread of HCI research into these issues is based on the idea that designers should try to match the strength/forcefulness of the highlighting techniques used with the importance/urgency of the message being delivered. That is, highlighting strength should be "proportional" to the utility that the user may gain [120, 74]. Gluck et al. conducted a study about the association between highlighting strength and information utility; although they failed to find any significant performance differences from matching strength and utility, they did find that users were less annoyed when the strength and utility were suitably aligned [74].

A convenient way of thinking about these issues is to consider the *importance and urgency* of the information being conveyed (Figure 2.5). That is, strong or disruptive techniques (requiring fast or immediate response) should only be considered when the information is urgent and important (i.e. top left quadrant). However, if the information is not important but still time critical (i.e. top right quadrant), it may still be a good idea to use a technique which will has a short response time, but one which is less distracting. When dealing with important but non-urgent information (i.e. bottom left quadrant), these can still be highlighted, but the techniques used should not cause distraction, and a less noticeable (i.e. time to notice the highlight is a bit longer, but it is still within a timeframe that means that the user will manage to notice it). In the last case (i.e. bottom right quadrant), highlighting techniques should be used sparingly, if they are used at all.

	Important	Not Important
Urgent	Interrupt NOW	Candidate for Interruption
Not Urgent	Aim for *awareness* but not *interruption*	Do not bother?

Figure 2.5: *Priority of information in terms of importance and urgency of communication.*

Design Insight 13
A convenient way of considering whether a highlight is needed (and how strong) is to consider the importance of the information being highlighted, and how urgent it is that the user is made aware of this information. See Figure 2.5 for details.

There are several issues here though, such as defining what information is important to the user, what information is urgently needed/useful for the user to know, and what does not fit these criteria. These issues can be best addressed by designers when considering the tasks, workflows, and mental models that users of their systems have [94]. For example, designers need to consider what goals are the users trying to achieve, how they aim to go about this, and what they may or may not know when doing this. Increasingly in mobile operating systems, the focus for this exercise needs to stretch beyond the app's own screens, and also across the various other places where the user can interact with it (such as the home/apps screen, notifications tray, or lock screen to name a few).

Another issue related to the strength of the effect is how long the interruption lasts. The Web Content Advisory Group recommends that disruptive elements (i.e. those with motion or sound) on web pages should not continue for longer than 5 seconds at a time, to allow the user to be able to focus on their tasks (e.g. reading an article) if they choose to not attend to the disruptive element immediately [4].

Design Insight 14
Disruptive elements (i.e. those with motion and sound) should not continue for more than 5 seconds, to allow the user to focus on their primary tasks [4].

2.2.4 Temporal Sensitivity

Temporal Sensitivity refers to our sensitivity to a stimulus that changes over time. Here, sensitivity refers to our ability to detect the stimulus [161]. It is useful for the HCI community to be familiar with this aspect of human perception, as many different types of popular highlighting techniques rely on these types of stimuli. Examples of such techniques include patterns of flashing or blinking [80], repetitive shaking or travelling motions [29], and kineticons [81].

The field of *psychophysics* in psychology has been particularly focussed on understanding how sensitive humans are to stimuli that change over time. A large number of very detailed studies have been conducted to study and characterise this aspect of human perception [161], covering a wide range of different factors that influence how sensitive we are to those stimuli.

Figure 2.6 shows the relationship between the perceived intensity and frequency of the effect. The perceived intensity is how sensitive our visual system is to such stimuli. Stimuli which are perceived as more intense are more likely to capture or redirect our attention towards them. The frequency is the rate at which the temporal changes occur, such as number of pulses per second or number of frames per second. These frequency thresholds here were identified and aggregated from various sources, including articles from Nasa's AMES Color Lab (for the most sensitive frequencies [15]), W3C's Web Content Accessibility Guidelines (in particular, the section on the seizure prevention threshold [2]), and various other sources for the other extreme thresholds [28, 159].

Several key observations about temporal sensitivity are shown in Figure 2.6.

1. **Temporal Sensitivity increases from 0 Hz to 8 Hz** – This suggests that at these "low" frequencies, highlights can be made more salient/forceful by using higher frequencies. It also suggests that highlighting techniques with a temporal component (i.e. when *Frequency* is non-zero) should be more noticeable than static (i.e. 0 Hz) techniques [161].

2. **Temporal Sensitivity peaks at 8 Hz, and falls off at higher frequencies** – There is no benefit to using any frequency higher than 8 Hz. [161]

3. **Humans are most sensitive to frequencies between 4 and 8 Hz** [15]

4. **Frequencies above 3 Hz can cause epileptic seizures** [2] – The use of higher frequencies (i.e. those above 3 Hz) carries the risk of inducing epileptic seizures in users [2]. These risks are accentuated with red stimuli that flash or blink (i.e. "red flash") [1]. Therefore, for interfaces targeted at a general audience (i.e. most interfaces, outside of those found in specialist equipment such as plane cockpits), it is generally recommended that frequencies higher than 3 Hz *should not* be used [2, 1, 159].

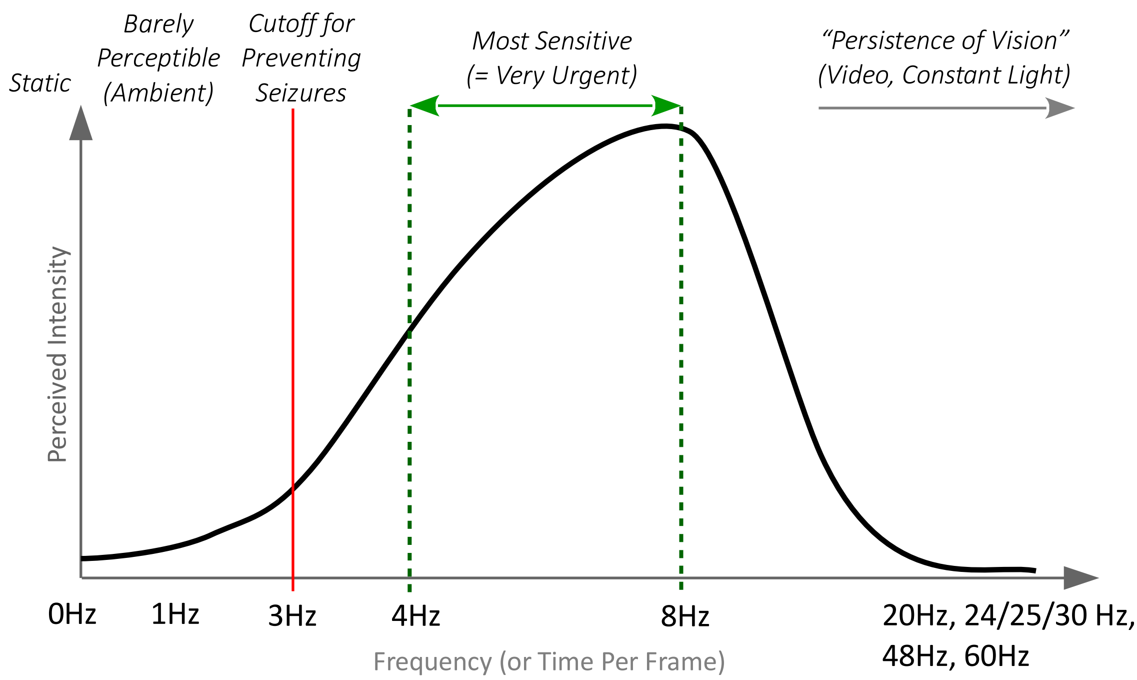


Figure 2.6: *Graph showing relationship between sensitivity to dynamic stimuli and the speed at which it changes. Not drawn to scale. Adapted from multiple data sources [159, 15, 2, 28].*

Design Insight 15
Highlighting techniques that change over time should be more noticeable than those that display a static/fixed state.

Design Insight 16
The safely usable range of temporal frequencies for highlighting techniques for most interfaces is 0 Hz to 3 Hz (inclusive). Higher frequencies within this range are more easily noticed than lower frequencies. Frequencies outside this range should generally be avoided, but may be able to used if the effect only pulses a few times.

Ware et al. [160] found some evidence that frequency or the speed at which an animation repeats may affect the perception of urgency, but Bartram [28] did not find any evidence of this within the 1-3 Hz range. Turner et al. [155] studied light flashing patterns for the lights on top of emergency service vehicles (e.g. police cars, ambulances, and fire trucks), and found that lights flashing at 4 Hz were considered to be more urgent than those flashing at 1 Hz.

Research Question 1
Does frequency have any effect on reaction time, accuracy, or distraction? Is there correlation between frequency and perception of urgency (or other similar semantics)?

An explanation for the drop off in temporal sensitivity at high frequencies is that when the temporal frequency is high enough, the human brain cannot effectively discriminate between the separate pulses [159]. This links back to principles of detecting signals from noisy inputs (from signal detection theory) [39] – namely, the idea that when incoming events are spaced closer together than the resolution of the temporal "sampling window" permits, they cannot be accurately discriminated (i.e. "fusion" occurs [161]). A practical example of this phenomenon in actions is how we perceive light sources and old CRT displays as being "constant light", even when they are actually constantly oscillating and flickering [159]. The "temporal sampling window" comes about as a result of limitations on how fast humans can perceive and process stimuli [44, 93].

2.2.5 Colour Perception

Colour plays an important role in many different highlighting techniques. Examples of common uses of colour include making items stand out, helping the user identify different groups of items, and communicating additional information about the items being highlighted. When used effectively, colour differences are a mechanism that designers can use to draw attention to information very quickly (i.e. by creating pop-out effects). Therefore, understanding how colour can be used effectively is important for designing effective highlighting techniques.

2.2.5.1 Luminance and Contrast

Luminance refers to the measurable amount of light that is reflected off a surface, while **brightness** refers to the *perceived* amount of light being emitted by an object [159]. **Luminance Contrast** refers to the difference (contrast) between the luminance of two colour patches [23].

In general, psychologists argue that the *contrast* between colours rather than the *absolute* colour intensity or the wavelengths/hues used is what matters the most when considering how well we can detect different coloured items [23, 161]. The colour usage guidelines provided by the Nasa AMES Research Centre Colour Laboratory claim that *luminance contrast* matters more than the absolute colours used [23, 20]. Watson claims that when considering temporal sensitivity for visual stimuli, the wavelength of light (i.e. colours) involved is mostly irrelevant, and that the intensity of the light used is more important [161].

> **Design Insight 17**
> *The luminance contrast between different colours matters more than the hue or absolute intensities of the colours used. Graphical elements are more legible (i.e. able to be distinguished) when there is sufficient contrast between them.*

If there is insufficient contrast between two colours, it does not matter if the hues of the colours are different. When there is insufficient contrast between the two colours, the difference between them falls below the "Just Noticeable Difference" (JND) threshold needed for identifying the presence of more than one colour [39].

2.2.5.2 Differentiation and Grouping using Colour

Apart from contrast, our ability to distinguish between different colours is also affected by a number of other factors such as the size of items (or thickness of lines), sharpness of edges/boundaries, the colour of the background/nearby areas, and also the number of different colours used in the display [16, 21].

The size of items has the greatest effect on how well we are able to detect different colours. Specifically, our ability to distinguish between different colours is poor when the colour patches are small (e.g. with thin 1-3px lines, or dots smaller than 3-5mm). This is perhaps familiar to many DIY home-decorators who have chosen a colour from a 1 inch sample on a colour swatch, only to discover that it appears completely different after painting it on a wall.

Design Insight 18
Users have difficulty discriminating between different colours when smaller colour patches are used (e.g. thin lines or dots smaller than 3-5 mm). Therefore, differences should not be solely communicated using colour when dealing with small items.

Not all objects with colour applied are solid shapes. Lines or curves present a number of additional challenges [22]. For example, animation tools present motion curves as a set of coloured lines [135], where different colours are used to distinguish between different curves, and one of the curves needs to be "highlighted" in some way to indicate that it is the curve that the user is editing. In these cases, these lines are often quite thin (typically 1-3 pixels thick, as thicker lines lose accuracy) and long (stretching across the screen). Line thickness is used as a measure of size here.

Studies have found that the number of colours (or more accurately, colour categories) that can be used in a display are limited [21]. For **normal sized objects**, there can be a **maximum of 6 to 10 colours**, while for **lines**, there is a **lower maximum of 4 to 5 colours** [21]. However, as the number of colour categories used increases, colour becomes less of a "unique" feature, thus reducing its salience as feature that pops-out to the observer.

Design Insight 19
There should be no more than 6 to 10 different colours used in a display with icon/widget sized objects.

Design Insight 20
There should be no more than 4 or 5 different colours used to differentiate lines. Up to 6 to 8 colours can be used, but the user will not cope as well.

Design Insight 21
*"**When everyone is special, nobody is special**": Using too many colours to group items in a display decreases the overall effectiveness of any highlighting techniques used (particularly those based on colour-based differentiation).*

Other issues that affect how accurately the user can identify the colour used include the background colour and sharpness of edges.

> **Design Insight 22**
> *Non-neutral background colours such as blue or purple alter the ways that we perceive the colours of items in the foreground, resulting in colour casts/shifts [16, 159].*

> **Design Insight 23**
> *Colour contrasts can be detected more accurately when there is a sharp edge between the colours [16].*

2.2.5.3 Communicating Values Using Colour

Colour can be used to communicate values such as "intensity" or "importance" by using different shades of the same colour, or by blending smoothly between several colours [16]. Different shades of the same colour are not subject to the "different colours" restrictions mentioned above.

While "rainbow" colour schemes (created by cycling through all the hues at full saturation) are often used for this purpose, this is problematic as not all hues have the same perceptual intensity (e.g. blue versus yellow at full saturation) [122]. These perception issues make it hard for the user to notice subtle differences in the coloured data. That is, **instead of highlighting important points, rainbow colour maps suppress key details**. Moreland's *Diverging Colour Maps* [122] solve these problems by using colours which keep the perceptual colour differences between different points along the colour map the same.

2.2.5.4 Other Design Principles for Using Colour

The Nasa AMES Colour Lab also published a few other useful guides about selecting colours [20], and how to design interfaces to make it easier for the user to interpret under various conditions [18, 17]. These guidelines were developed for designing clearer flight instruments that were less error prone, where the user needs to view the displays under varying lighting conditions, and when they are subject to potentially high workloads in stressful situations.

2.2.6 Human Performance Thresholds

Human movements and cognitive processes take time to perform. For instance, it is physically impossible to instantaneously move our eyes, decide how to respond to an event, or press a key/button [39]. In Table 2.1, we list a few of these lower bounds (i.e. the minimum/average time required for these actions to occur) [94, 44].

Item No.	Action	Time Required (seconds)
1	Visual fusion	0.05
2	Lag before full awareness of a visual event	0.1
3	Delay after awareness of event before the eye fixates on the item [85]	0.1 - 0.25
4	Time to identify items	0.25
5	Limit on perception of cause and effect	0.4
6	Minimum visual-motor reaction time (after awareness)	0.7

Table 2.1: *Minimum/average times needed for the human body to perform several operations of importance when detecting highlights.*

Each of these times are important for the following reasons:

- **Visual fusion** – As discussed in the Temporal Sensitivity section (Section 2.2.4), visual fusion occurs when we cannot distinguish the frame-to-frame differences of fast changing items.
- **Lag before full awareness of a visual event** – Before we are fully aware of a visual stimulus, we may not be able to process it.
- **Limit on perception of cause and effect** – If a highlighting technique is intended to communicate to the user that performing a particular action (i.e. selecting a file/folder) corresponds to some other location/tool, the highlight must appear within 0.4 seconds or else the user will not be able to associate the highlight with the action which caused it to appear.
- **Minimum visual-motor reaction time** – Even if the user is able to quickly detect a stimulus, the user will still not be able to start performing a movement to react/respond. The main implication of this delay is that *experimental methods relying on reaction time may not have sufficient sensitivity* to detect detection times of less than 0.7 seconds. This may be one of the contributing factors which lead to performance ceilings/floors being identified.

2.3 The Role of Semantics

Although humans have an innate ability to perceive and recognise visual stimuli, we need to learn how to interpret these stimuli to make sense of the world around us. An example of this is learning how to interpret certain patterns of glyphs as letters and words, and also the sounds and meanings of those. Another example is learning what certain icons (e.g. flags, emoticons, abstract symbols such as the "play" triangle), shapes (i.e. silhouettes, sharp versus rounded edges), and colours are used to represent. Finally, there is the issue of inferring emotions or physicality from moving images.

Cultural differences (such as different interpretations, practices, and/or belief structures) mean that people from different cultures and different parts of the world associate different meanings with different types of stimuli. These are all learned/passed on and affect the way that we view the world. Such biases play out at the individual and demographic level [55, 123, 136], resulting in substantial variations in aesthetic judgement as a result of this [108, 115].

Semantics can also apply across cultures. For instance, the Gestalt Principles [102] are a set of graphic design principles that are used by designers to group items together, communicating whether item A is related to items B, C, and D, or whether item E is distinct from item F. Another example of how semantics can be cross-cultural is the way that physics of the natural world affect the way that we interpret images. For example, we infer shadows, illuminated surfaces, and "glowing" light sources in two dimensional images from our experience with how light behaves in the real world [9].

The study of semantics is a broad topic. In this section, we identify some of the semantics most relevant for highlighting techniques. First, we discuss the Gestalt Principles and how they can be used to highlight items by putting them in separate groups from the set of non-highlighted items. Second, we discuss the role of semantics derived from the way that the physical world works – such as how illumination, shading, and motion can be used to imbue meaning to visual elements. Finally, we dicuss how some cultural factors – such as the aesthetic appeal of colour and complexity, the semantics of the colour and shapes used – affect how different visual effects/designs are interpreted by users.

2.3.1 Semantics of Gestalt Principles

The Gestalt Principles of visual appeal are a well-known set of graphic design guidelines [102]. The principles most useful for highlighting (i.e. for showing whether an item belongs to a set or not) are [159]:

1. **Connectedness** – Items with physical links between them appear to be part of the same entity.
2. **Proximity** – Items spaced more closely together are perceived as being in the same group.
3. **Similarity** – Items with similar *colours*, *sizes*, or *shapes* appear to be related. For example, to communicate whether certain icons are tools or options, the icons may be colour coded (e.g. orange for tools, blue for options/settings). Another example is how icons for file-types and file operations often feature a "page" icon.

As can be seen from Figure 2.2, there are similarities between the "visual features" that the visual system has dedicated visual processing pipelines for [154], and the visual manipulations that can be used to communicate that groups of items are related [102, 159]. This implies that highlighting techniques which need to create "pop-out effects" (i.e. they are rapidly noticed by the pre-attentive visual system) should use unique visual features not used elsewhere in the interface.

Design Insight 24
The Gestalt Principles can be used to create pop-out effects, by making highlighted items appear to belong to a different group to non-highlighted items on a pre-attentive level.

Typically, highlighting techniques attempt to boost the visual salience of target items. Examples of techniques used to boost salience include increasing the saturation or brightness of colours, increasing line thickness (or size of items), or adding additional elements (e.g. the visual elements in the *Underlay/Overlay* layers in Figure 4.1).

However, sometimes it is necessary to instead *suppress* the salience of items not in the target set, as the target items cannot be made more visually salient. This situation could arise when the target/highlighted items already use the brightest colours possible without impairing the observer ability to discriminate between different hues. For example, Bergman et al. [34] developed a file-browser interface where all the non-highlighted items were greyed out (i.e. saturation was reduced) while leaving the highlighted ones unaffected (i.e. their saturation is untouched, or subtly boosted where possible).

Design Insight 25
Items can be highlighted by boosting the visual salience of the item being highlighted, or by making the non-highlighted items less salient so that the highlighted item stands out more.

2.3.2 Semantics of Illumination and Shading

Illumination and shading effects can be used to impart meaning on visual objects. For instance, shadows can be added around/behind an element to suggest that it it stacked above the items around it, making it appear to protrude out of the screen. This simple trick was used in early UI's to give buttons a "clickable" affordance [126].

Similarly, gradients and other types of shading effects can imply that an element is a curved, angular, or rugged surface. Shading/illumination effects can also be used to direct attention to at important items.

2.3.3 Semantics from Motion and Dynamic Effects

Motion and dynamic effects (such as colour changes and distortion) can be used to convey meaning and emotion. According to Williams [174], humans are able to make complex inferences about the movements of a simple abstract shape. This is possible when inanimate object behave in a physically plausible ways according to the laws of physics [104].

Chang and Ungar [47] argued that interfaces should make greater use of animation, to appear more lively and natural. They claimed that judicious use of animation could help lower the user's cognitive load, by relying on the motion semantics to communicate attributes about a visual element (e.g. that a window was related to a particular icon, because it expanded-out from that icon).

In a taxonomy of UI motion effects, Harrison et al. [81], identified several categories of motions based on their origin: *biological motion* (i.e. human and animal motion), *gestures* (e.g. nodding, shoulder shrugs, or thumbs up), *organic motions* (e.g. the beating of a heart), mechanical motions (i.e. the movement of buttons, knobs, and toggles), the *effects of physics* (e.g. leaves blowing in the wind), and *cartoon conventions* (e.g. squash and stretch effects). From this taxonomy, Harrison et al. developed a proof-of-concept set of 39 *kineticons* (i.e. animation effects applied to UI elements by transforming/deforming them) [81]. The results revealed that there were strong associations between certain kineticons and particular concepts/meanings, with 36 of the 39 effects studied being "significantly more likely to convey certain meanings than others" [81].

Another study by Harrison et al. [80] performed a detailed structured-design investigation into a design vocabulary for controlling the intensity of LED lights (or "point light sources") to create different patterns of flashing lights. Such indicators are commonly found on mobile devices, which have a LED indicator (commonly found in the top left corner) that can be controlled by different apps and/or the operating system to communicate information to users. Harrison et al. concluded that the existing design space had not been adequately explored, and that the vocabulary of designs in use was "small, fairly unimaginative, and generally ambiguous in meaning" [80]. After conducting a structured design process with input from various designers and a series of exploratory user studies, they identified a set of eight different effects which were particularly effective at communicating common device states. For example, they recommended using the "Beacon", "Bright Flash", "On with Bright Flash" effects for notifications [80]; they also recommended using the "Pulse Slow", "Fast In Slow Out" for indicating low-battery status [80].

2.3.4 Semantics of Cultural Factors

Cultural differences in the way that different stimuli are interpreted has effects on the suitability of using certain highlighting techniques in certain environments. In this section, we cover the three most relevant ones: *Colour*, *Shape*, and *Aesthetic Appeal*.

2.3.4.1 Semantics of Colour

Colour has semantics associated with it including trustworthiness, importance, and urgency [107, 159, 138]. For example, red, yellow, and green are often associated with safety [21].

Design Insight 26
Colour is often associated with certain concepts. By respecting these conventions, designers can improve the speed and accuracy with which the user can understand the message conveyed by a highlighting technique.

2.3.4.2 Semantics of Shape

Interpretation of shape may vary across cultures [37]. For example, Kim and Lee [100] found that cultural differences can influence how well people from different cultures are able to recognise icons using abstract versus concrete designs. Another example is how different languages use different combinations of glyphs to communicate similar concepts.

However, shapes (in the form of icons) can also be used to communicate concepts in a cross-cultural way [81] (e.g. the "Floppy Disk" icon used to indicate the *Save* operation, or a right-facing isoceles triangle representing the *Play* operation).

> **Design Insight 27**
> *When manipulating object shapes to create highlighting effects, it is important to consider cultural differences affecting how they are interpreted (e.g. the ease of understanding the implied concepts, and how symbols can have different meanings associated with them).*

2.3.4.3 Semantics of Aesthetic Appeal

Reinecke and Gajos [137] demonstrated that cultural factors strongly affect aesthetic assessments. Their study focussed on two primary factors: colourfulness and complexity. Colourfulness was determined using a method based on Hasler and Susstrunk's work on computing perceptual measures of how colourful RGB colours viewers perceived colours to be [82, 138]. Visual complexity was measured by using a regular grid pattern and a quadtree [138]. Participants in this study (recruited from across the world using a crowd-sourced "online laboratory") were asked to rate snapshots of a series of pre-selected websites, with the resulting user-rankings compared against the colourfulness and complexity metrics, along with a few other similar candidate measures of aesthetic appeal.

They found that the level of colourfulness and visual complexity for attaining the highest appeal among different audiences varies [137]. *Colourful* (in particular, highly saturated) designs were preferred more by female viewers, less by those with a higher education level, and the most by Macedonians. *Complex* visual designs were most likely to be preferred by Asian viewers, while Russian and Finnish viewers were most likely to prefer simple designs.

Their findings have implications for the use of highlighting techniques, especially regarding the subjective preferences of users from difference demographics towards or against certain techniques. For example, divergence from the expected cultural norms for visual complexity that a user is accustomed to could affect their performance (e.g. a user accustomed to low information densities may struggle when faced with high density displays featuring multiple highlights competing for their attention).

> **Design Insight 28**
> *Users may have more difficulty noticing highlights in a visually complex interface if they are normally only exposed to interfaces with a low information density. However, high strength highlighting techniques may be less affected by this problem.*

It is also possible that there may be a link between how "annoying" or "ugly" a stimulus is perceived to be, and how distracting its effects are to users. The user's distaste for a stimulus may lead to a heightened awareness of its presence, leading to negative effects on performance (i.e. the user becomes distracted by the annoying/ugly stimuli).

Research Question 2
Are stimuli perceived to be "annoying" or "ugly" also more distracting for users?

2.4 Summary of Human Factors

In this chapter, we reviewed underlying human factors which affect how users interact with highlighting techniques.

We discussed how humans are only able to focus on a very small part of the world at any time. The structure of our eyes means that most of the colour sensitive cones are located in a small region (fovea) the middle of our retinas/visual field. Vision in the fovea is sharp, while our ability to perceive colour or details is impaired across the rest of our visual field (peripheral vision) [159]. That is, items that fall further away from the foveal region are less able to be detected, especially if they occur in a cluttered area (where items are closely spaced together). Our limited cognitive resources mean that attention is a very scarce resource that can only be directed towards a single thought/task/target at any time [95].

Our executive control system (in the prefrontal cortex) facilities determine what we focus our attention on, and where we direct our eyes to look (i.e. what do we want/need to see in detail by directing our foveas towards that region) [159]. Sometimes this may come at the cost of ignoring stimuli that were vying for our attention – either deliberately, or accidentally. Change Blindness [139] and Inattention Blindness [113] are two phenomena where we fail to notice highlights or changes due to external factors (i.e. "visual disruptions" for CB), and internal factors (i.e. our attention was occupied by another task/point of focus, or IB). To prioritise where their attention should be directed, users develop a mental model of the relative information utility provided by different parts of the UI (i.e. how frequency there is useful or important information to attend to, relative to the cost of missing the event) [159, 163], and use this to inform their behaviour (e.g. how alert they are, and the scanning patterns used to compensate for CB and IB) [131].

We also discussed factors which may influence how well the user can detect various types of highlighting effects. Treisman and Gelade's theory of how visual search works [154] suggests that pop-out effects (which allow the user to very quickly identify a highlighted item) can be created by using a unique graphical feature. The semantics associated with different types of stimuli [37, 102, 159, 81] affect the way that we interpret and react to them. Studies of temporal sensitivity (or how well we are able to detect stimuli which change over time) reveal that the usable range of frequencies for repeating animations are 0 to 3 Hz, with higher frequencies being more noticeable (i.e. static elements are less salient than moving ones) [161, 2]. The studies of temporal sensitivity also reveal many interactions with other factors, such as how larger items are more easily noticeable at low frequencies, and how brighter backgrounds make it easier to detect moving stimuli [161].

3 Previous Studies of Interactive Highlighting Techniques

This chapter reviews studies of highlighting techniques and methods for measuring, comparing, and analysing their effects in the HCI literature.

3.1 Example Applications of Highlighting Techniques

In this section, we discuss a few examples of different applications for highlighting techniques in HCI. There are more applications than the ones discussed here. The intention is to illustrate the breadth of possibilities, and to reframe some existing work in the context of highlighting techniques in the hope that the HCI community can unlock some new and previously overlooked possibilities.

3.1.1 Notifications

A common use case for highlighting techniques is to implement notifications and status indicators. Examples of these include the "Downloads Complete" status indicator in Firefox, the Picasa "Star" (for indicating an important/good photo), or "Alert Bars" (e.g. the yellow overlay that appears across the top of a webpage when popup windows were blocked by the browser).

The design of traditional widgets – for example, the way that buttons being "pressed" appear recessed – were aimed at projecting the affordance that these were elements that could be interacted with [126], and that by looking at their visual design, it would be possible to identify what their current state was. This in turn was often exploited by user interface designers. For example, the Bold/Italic/Underline toggles found in most word processor interfaces are examples of toggle buttons used to convey state information at a glance.

There are a number of examples of this in commonly available software. For example, when downloads are completed, Mozilla Firefox shows an animation where a large green arrow (i.e. basically, a larger version of the icon used to indicate completed downloads) is spawned from the downloads button, growing larger and fainter over the course of a 1-2 seconds. Once the animation completes though, all that's left is a green icon on the downloads button (which has a grey icon by default). An equivalent technique is also used in the Chrome web browser.

Alert bars are a type of widget seen in Gnome applications as well as web browsers (e.g. the "popups blocked" message in Firefox, and "automatic translation" offer in Chrome) which are used to display messages to users in a non-blocking way by displaying them in a banner that runs across the top of the interface (just under all the other standard toolbar chrome). An animated transition (often, a sliding-downwards effect) is used in some implementations when starting to show these widgets. However, once visible, the technique relies on its static saliency (i.e. primarily colour, size, and to a lesser extent position) to attract attention. As with the downloads completed animations, these do not disappear without user intervention.

Anderson et al. [12] investigated how different highlighting techniques could be used to make the user pay attention to security alert dialogs instead of ignoring them (as with many other popup dialogs [103]). Specifically, they were investigating the efficacy of "polymorphic warnings" – that is, dialogs which randomly use different highlighting techniques each time they appear, so that the user does not begin treating the appearance of a particular type of stimulus as part of a routine (and/or become desensitised to a particular type of stimulus).

3.1.2 Scented Widgets

"Scented Widgets" is a term coined by Willet et al. [173] to describe the concept of adding additional visual hints to widgets to provide additional information scent [131]. Most of the techniques described in the original paper are static (i.e. non-animated). Note however that although the decorations and hints added to widgets are non-animated, they can still change in response to user actions. For example, one of the scented widget techniques displayed usage-frequency histograms alongside sliders; although these histograms were not animated, their distributions dynamically changed in response to user input.

Chen et al. [48] extended upon this work to develop scented widget techniques for helping to guide the user through the process of filling out of form. Agapie et al. [10] also investigated a "scented widget" approach for input validation by changing the colour of a textbox to indicate whether the input was acceptable. Atwood et al. [25] reported various attempts at using highlighting techniques for input validation of a longer-form textbox.

Other examples of prior techniques which have similar characteristics include the Readwear [84] and Footprints [11] scrollbars, which also draw information scent hints on the widgets themselves to indicate useful locations. A modern instantiation of this technique in commercially available software is the search highlighting in Chrome [77].

3.1.3 Managing Information Overload

Highlighting techniques can be used to prevent information overload by helping to direct user attention towards items that they are likely to be interested in (for example, based on prior behaviour). Highlights may achieve this effect by acting as landmarks in the information space. *Landmarks* are traditionally used in real-world navigation to help people orientate themselves in an environment. Although the mechanisms are not yet well understood, landmarks appear to play an important role in helping people develop their spatial memory of an environment [143]. In theory, anything in an environment may be able to serve as a landmark. Intuitively though, effective highlights need to be salient in some way by being particularly eye-catching, holding some significance to the user, or having some unique features relative to the surrounding environment (e.g. special sounds or smells). This is because landmarks would need to be quickly (if not instantly) identifiable, in order to be of use to the user when navigating.

There is also the problem of what is relevant to highlight as a landmark. From moment to moment, the user's needs change as they progress through their tasks. To support these changing needs, the highlights shown need to also change and adapt to the situation (i.e. "dynamic landmarks" [132]).

3.1.3.1 Navigation Hints in Large Information Spaces

Navigation in large information spaces such as hierarchical file systems can be an inefficient and cumbersome process [36, 33, 35, 66]. Early attempts in HCI to solve these issues centered around creating new types of visualising the information space (e.g. TreeMaps [92] and Fisheye Distortions [71]).

Recent attempts however have focussed instead on making it easier for the user to identify what they are looking for [66]. Fitchett et al. [67] developed the *Icon Highlights* (IH) and *Search Directed Navigation* (SDN) techniques for helping the user identify files they had recently accessed (IH), or to help the user figure out which sequence of folders they needed to navigate through to find the files they needed (SDN). These techniques worked by drawing circular halos around highlighted items, and/or drawing a gray overlay over other non-highlighted items (i.e. "suppression"). Bergman et al.'s "Old N' Gray" technique made older/less frequently accessed items less salient [34].

Other examples of highlighting techniques used to improve navigation interfaces include scrollbars enriched with revisitation hints such as Read Wear scrollbars [84] (where the past frequency of revisitation of each section in a document would be drawn on a corresponding part of the scrollbar), the Footprints scrollbar [11] (where coloured and numbered squares were placed on the scrollbar, at locations that the user recently visited), and Chimera's Value Bars [50] for showing the locations of items with similar attributes.

There have also been studies on more fundamental issues such as how many highlights or landmarks can be shown to the user while still improving performance. Quinn et al. [132] performed an experiment to determine the optimal proportion of landmarks that could be presented to the user. The results of their study suggested that it was best to use a logarithmic function (i.e. $f(n) = \log_2(n)$, where n is the total number of potential landmarks that could be shown) as the "*Dynamic Landmarking Function*" used for determining the number of landmarks to present to the user [132].

3.1.3.2 Subconscious Navigation Hints

While menus may not be as large and complex as a file system, there are also similar issues here when users encounter a large menu containing many infrequently used items.

Ephemeral Adaptation is a technique that was developed to solve this problem [64].

Yantis and Jonides [176] demonstrated that items that appear abruptly or suddenly are noticed quickly (and without requiring much conscious effort), whereas items that gradually appear take longer to be detected. They also found that items with abrupt onset were able to be detected quickly even when users had not been primed to search/look for such criteria, as is necessary for other pre-attentive features such as colour [64]

Design Insight 29
Items that appear abruptly or suddenly are noticed quicker (and with minimal conscious effort), whereas items that gradually appear take longer to be detected.

This is the psychological basis used for the *Ephemeral Adaptation* technique [64]. The highlighted set of items appear immediately (i.e. abrupt onset), while the non-highlighted items gradually fade in over 250-1000 ms (i.e. gradual onset). In other words, the Ephemeral Adaptation (EA) technique is an example of a suppression-based technique, where saliency of non-highlighted items is reduced (i.e. by having these gradually fade in, to be less likely to be noticed, and to provide an opportunity for the user to notice the highlighted items first). It was found that EA was significantly more effective at helping users identify the target items in a menu than static highlighting (i.e. shaded background fills), and that adaptive highlighting was more effective when used in conjunction with EA than without.

3.1.3.3 Communicating Spatial Correspondence

Spatial correspondence techniques are used to show how different perspectives and parts of an information space are related to each other.

Gliimpse [60] is an interface designed to help the user understand how code written in markup languages such as HTML, LaTeX, or Markdown corresponds to the final compiled output. Instead of presenting the user with two separate views (e.g. code on the left-hand side, and PDF output on the right-hand side [42]), it uses a single-view interface where the user can switch between the code and markup views of the document by pressing a hotkey. Smooth animated transitions are used to show how commands in the markup code correspond with section headings and input fields in the translated output. For example, the transition for an HTML form has 4 stages (see Figure 2 of [60]):

1. The unaltered HTML code;
2. The form with the basic layout, basic fonts, and the code-blocks for the input fields shown with blue shading;
3. The form with proper fonts, and shaded blocks where the fields will be; and
4. The final form layout.

Diffamation [49] is a related technique to help the user visualise the changes made from two versions of a given document. The name of this technique is derived from the term, "diff", which is also the name of a tool used by programmers and other advanced users [98] to identify the line-by-line differences between two versions of a file to identify the changes made.

The system consists of three related components – a timeline, scrollbar, and text viewer – which together form the basis for the Diffamation interface. Unlike their traditional equivalents, the scrollbar and text viewer components show highlights (e.g. red/pink highlights for removed text, and green for added) for the components which are changing, with animated transitions used to blend between the different states [49]. Using the timeline, users can navigate through the change history of the file, and watch as different sections dynamically appear and disappear, with smooth transitions between these [49].

3.1.4 Novice to Expert Transitions

Closely related to the work on file retrieval interfaces is the issue of facilitating smoother transitions from novice to expert performance. Highlighting techniques are particularly promising approaches for doing this in a less intrusive way than forcing the user to read a manual or tutorial document first.

3.1.4.1 Teaching Mechanisms and Tours

Grossman et al. tried using different highlighting techniques to encourage users to learn hotkeys [78]. However, they found that despite trying various measures to redirect the user's attention towards the hotkey information, the most effective technique was still to force users to use the hotkeys by preventing the tools from working when invoked from the menus via point and click.

Tip of the day features and guided tours are commonly used to teach users new functionality. Guided tours typically feature popouts (similar to the one shown in Figure 3.1 situated alongside newly added UI elements (or on existing elements used to access the new feature).

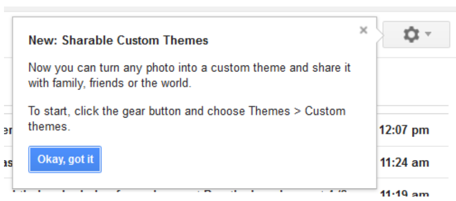


Figure 3.1: "New feature" callout balloon used by Google to illustrate a feature newly introduced to Gmail

Another class of approaches is to use shading to indicate the utility of different techniques (similar to Willet et al.'s "scented widgets" idea [173]). Examples of these approaches include Grossman et al.'s *Patina* [117] (which shows a heatmap of aggregated usage statistics of how often different functionality is used), and Scarr et al.'s *StencilMaps* technique [143].

Scented widget techniques can also be used to teach the user hotkeys. Malacria et al.'s *ExposeHK* [112] overlays hotkey labels over toolbar buttons and menu items when an accelerator key (e.g. ALT or CMD) is pressed. Giannisakis et al's *IconHK* [73] embeds and blends hotkey indicators into toolbar icons to increase the accessibility of keyboard shortcuts.

Autodesk Maya contains a tool which can be used to highlight all newly added features in a given release [26]. This is done in several ways, ranging from applying green shading to the background of new menu items, to drawing green boxes/borders around toolbar icons, buttons, and other options. Unlike the Windows Start Menu example, this is an example of an "on-demand" help feature, which must be explicitly activated by users on an as-needed basis (whereas the Start Menu feature is enabled by default, but can be disabled if users find it too distracting). A limitation of this approach though is that users must first know of the existence of this tool and also to consider it useful or likely to provide useful insights (as per Information Foraging Theory [131]) to consider investing time and effort using it.

Mac OS X has a similar feature built-in which can be used to show users where the tools they are trying to find are located in the menus/toolbars. The "oscillating blue arrow" technique (see Figure 3.2) draws a dark blue arrow that slowly moves in a circle beside the appropriate menu entry.

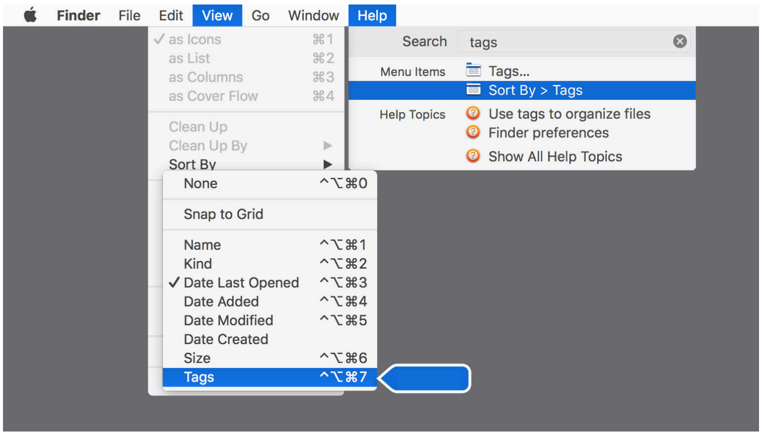


Figure 3.2: *The oscillating blue arrow feature in Mac OS X for showing users where the tools they are looking for are located in the menus. The "oscillating blue arrow" is the blue tag shape shown beside the 'Tags' menu item.*

3.1.4.2 Indicating New/Untouched Items

By default, the Windows Start Menu highlights newly added/installed items by shading these and their parent items in yellow/ochre colours. These highlights only disappear after the user has used the highlighted item. This is an example of a feature that is aimed at improving the novice to expert transition, by showing users functionality that they may not be aware of – specifically, "newly added" functionality – in a non-obstructive way that does not require immediate attention (in accordance with the nature of this sort of design goal).

It is most effective if users immediately attend to (i.e. investigate the cause of) the notifications – for instance, immediately after installing a new piece of software, as the cause-effect connection between stimulus (i.e. the highlights) and cause (i.e. an application was installed) is clear. This association becomes less clear though if a user does not do so, and these notifications are left to linger. In that case, it can be difficult to determine whether any new items were added (notably since the top-level "All Programs" item would already have been highlighted), and/or where the newest items are (since all the previously added but untouched items are also highlighted). From this example, we can see several of the problems inherent in any attempts to make users aware of functionality they may not be aware of: 1) *Temporal Relevancy* (i.e. only showing information that users are likely to want to know and/or respond to *now*, given their likely tasks/goals), and 2) *Avoiding saturation*.

3.1.5 Ambient Awareness

Ambient notification techniques are used to display some non-essential status information in the periphery of a user's field of attention [124]. Most of the techniques described by Mueller et al. [124] and Tomitsch et al. [153] are physical artifacts or displays which are placed near a user's monitor which undergo some form of crude change when the status they represent changes.

Examples of such techniques include strips of multi-coloured LED arrays mounted along the edges on the back side of a monitor, with the colours emitted by the LED's changing when an email or similar alert arrives and needs the user's attention [124]. Other examples include various USB-powered gadgets that sit on user's desktops and provide various status alerts using some combination of flashing lights, movement, and/or physical indicators [153].

For our purposes, these and related techniques involving physically-separated secondary displays (such as Matejka et al.'s "Ambient Help" system [116]) are out of the scope of our work, where we are only interested in techniques for use on the primary display (and primary focus of user's attention).

3.2 Methodologies for Empirically Measuring Noticeability and Distraction

This section reviews prior methods in the literature for measuring the noticeability and/or distraction effects of highlighting techniques. It provides important insights for understanding *how* noticeability and distraction can be measured, for identifying the *merits* (i.e. benefits and limitations) of prior methods, and for identifying *promising techniques/directions* for future research.

In the literature, we identified three main paradigms for measuring the noticeability and/or distraction effects of highlighting techniques:

1. **Dual-Task** – Participants are asked to concurrently perform two tasks: 1) A "Primary Task" (e.g. playing a game or some other "work-like" activity), and 2) A "Secondary Task" (i.e. detecting and responding the appearance of a highlighted item).

2. **Path Deviations** – Participants are asked to quickly point to the target from a grid of candidate items. During the analysis phases, the pointing trajectories are analysed to measure the extent to which the shape of the paths was affected by highlights.
3. **Short-Exposure Present/Absent** – Participants are briefly shown a set of stimuli, and are asked whether they saw a highlighted item or not.

3.2.1 Dual-Task Paradigm

Dual-Task experiment methods were used in many prior studies of highlighting effects in the HCI literature, as the dual-task setup is a good simulation of one of the main use cases for highlighting techniques – that is, the use of highlighting to attract and redirect user attention away from their current task/activity to focus on a piece of new information. Common examples of highlighting techniques used in such scenarios include "New Message" notifications, indicating state changes in the system, and teaching the user shortcuts.

Primary Tasks in Dual-Task studies typically require participants to perform some activity where they need to be fully engaged in the task at hand. Most tasks in the literature can be classified into the following categories (listed from most artificial to most realistic):

- **Dot-Following Tasks** – Participants use a mouse or joystick to track the movements of a moving target as accurately as possible [99, 86].
- **Work-Like Activities** – Participants perform an information gathering task (e.g. performing a hypertext browsing task [118]), document transcription task [160], or a dashboard monitoring task (e.g. air traffic control [86]).
- **Interactive Games** – Participants play simple games, where they have intrinsic motivation to concentrate on the task (e.g. to get a high score, in order to receive performance-linked prizes at the end of the experiment). Examples of games used included Solitare [29], and car racing games [57].

Secondary Tasks are dedicated to understanding how noticeable/understandable the highlighting techniques being studied are. Depending on the intended research insights, there are several variants used in the literature:

- **Time to First Noticed** – Participants are asked to press a button (or point to the highlighted item) as soon as they notice it [29, 86]. This is a direct measure of noticeability.
- **Response to Highlighted Information** – Participants are presented some information in a highlighted display, and are provided with an opportunity to demonstrate whether they noticed (and/or internalised the associated information) by acting on that information. For example, in Davies and Beeharee's study [57], participants playing a racing game were given hints which could improve their score; the objective of the experiment was to see which of these hints (using different highlighting techniques) were more effective at helping participants understand the associated information..

- **Accuracy of Detection/Response** – Instead of only measuring how quickly participants can respond, this approach is more concerned with how accurately participants are able to identify the highlighted item. Examples of this include the "Friend-or-Foe" classification task used in studies simulating radar controller workstations [99].

The main challenge that experimenters need to address when designing Dual-Task studies is to choose a suitable primary task. There are several tensions/objectives which need to be balanced when selecting a task:

1. **Measureability and External Validity** – It should be possible to easily and accurately determine how well participants are performing the task (i.e. *Measureability*). Some prior studies rejected the use of performance-based measures of distraction, noting that it was difficult to determine task performance (e.g. "What is a 'better' Solitare strategy" [29]). However, this sometimes conflicts with the desire that the task is a good proxy for the real-world activities of users (i.e. *External Validity*).
2. **Task Engagement and Difficulty** – The task must be sufficiently difficult to keep participants focussed on it, without being too difficult for participants to achieve satisfactory/consistent levels of performance.

3.2.2 Path Deviations Paradigm

The "Path Deviations" paradigm is a relatively new approach introduced by Gallivan and Chapman [72] and developed further by Moher et al. [121]. In this class of approaches, participants only have to perform a single task: pointing towards the target item, as quickly as possible, starting from a standard reference point. The pointing trajectory data is then processed and analysed to determine the extent to which participants subconsciously drifted towards a highlighted item, with a larger deviation indicating a stronger degree of distraction.

Pointing trajectories were measured as three-dimensional paths using an infrared marker attached to the participant's index finger [72]. At the start of each trial, the experimenters would carefully reset the participant's hand to a fixed reference position facing the vertically-orientated touchscreen. During trials, participants would fixation on a white cross in the center of the screen until the stimuli appeared. When participants had detected the target, they would move their hand to touch the appropriate spot on the screen. The path data was analysed by discarding the depth-component, normalising/resampling the paths to have the same number of points, and computing a *Distractor Attraction Score* to compare whether each point in the distractor-present trials were closer/further-away from the corresponding point in the reference (distractor-absent) trials [121].

Stimulus exposure times were time limited in one of two ways [72]:

1. **Forced Choice** – Stimuli were displayed for a short period of time before disappearing. Exposure times ranged 325 ms [72] to 1 second [121].
2. **Free Choice** – Stimuli were displayed until participants responded. However, the overall length of each trial still had time constraints (e.g. 500 ms [72]).

However, there are several limitations on the practicality of using these protocols. For instance, both studies had "strict acceptance criteria" for how trials were to be performed (i.e. each trial had to be performed within a particular timeframe to be considered a valid datapoint, and participants had hold a particular pose before the start of each trial). As a result, trials often to be repeated many times to obtain usable data.

3.2.3 Short-Exposure Present/Absent Paradigm

There are also other possibilities such as protocols where stimuli are shown to participants for very brief periods of time, and then they are asked to respond whether they saw a highlighted item or not. Gutwin et al. [79] conducted a study where participants were shown a field of 104-105 items for 240ms, and were asked to report whether they noticed the presence of the highlighted item at that time.

Variations on this basic theme may include asking participants to state how confident they were in their assessments, or asking participants to report some information associated with the "target" item. However, such approaches were aimed at only measuring the noticeability or detectability of different visual stimuli; as such, they were less suited to measuring/quantifying distraction.

3.3 Studies of Highlighting Technique Effectiveness

There are many examples of studies comparing the effects of different highlighting techniques to provide guidance to designers on the relative effectiveness of different techniques. Dual-task experimental methodology is often used, as highlighting techniques are often used to attract the user's attention when they are engaged in a task. This section identifies and documents several of the most common types of studies documented in the literature.

3.3.1 Effectiveness of Abstract Stimuli/Icons

One of the most commmon types of studies of highlighting techniques are those focussed on measuring the effectiveness of highlighting techniques using elemental graphical manipulation techniques (e.g. similar to those described in the Gestalt Principles [102] or by Bertin [37, 45]). Many of these studies may have been partially motivated by the belief that by measuring the effects of each of these elemental manipulations, there will eventually be sufficient data to construct a model of highlighting effectivness (or rather, human responses to them), as most highlighting techniques could theoretically be described in terms of these manipulations.

Bartram conducted several studies about the effectiveness of different highlighting techniques [28]. From these studies, Bartram identified a number of design guidelines for designers:

- **Motion can be used in combination with other techniques** such as colour, shape, and other ways of coding information visually. This is because it acts as an additional channel of communication, which does not interfere with other techniques.
- **Small repeating effects are more effective than static graphics**, especially in peripheral vision (where the majority of an interface will be seen by users when they are working)
- **Even motion amplitudes of 1 degree of visual angle are sufficient for highly detectable stimuli**. Detection accuracy is not related to the amplitude of movements. However, smaller amplitudes may lead to slower response times.
- **Even slow frequencies (1-3 Hz) are effective** (for repetitive motions)
- **Smoothness of motion does not matter**. This means that stuttering of playback is acceptable.
- **Level of task engagement/focus affects highlight detection performance**

Bartram found that "*anchored motions*" (those where the *Base* and *Content* layers (see Chapter 4 for more details about these concepts) have a transform which makes it oscillate around its midpoint) were significantly less distracting/annoying than "*travelling motions*" (e.g. circling motions like the Mac OS blue oscillating help arrow, or an element animated to move between two point using a saw-shaped F-Curves). "*Popping motions*" (i.e. effects where the item zooms in and out along the depth axis) were also considered distracting [28].

Bartram recommended that the best general purpose technique was "*slow linear oscillation*", as it provides good response times, can be accurately detected, and is non-intrusive and non-disruptive [28].

Kieras and Hornof developed a predictive model of active visual search performance [99]. They found that as eccentricity increases, the probability of noticing items differentiated using colour falls off slower than for shape or size (see Figure 7 from their paper). Specifically, while colour falls off gradually from 1-7 degrees of visual angle, both size and shape abruptly drop to zero around 3 degrees of visual angle. This could be explained by some of the work on crowding and clutter (see Section 2.2.2), such as the summary statistics model proposed by Rosenholtz et al. [141] (where the brain does not actually know or care about the details of the contents of peripheral vision, but rather, it uses a simplified representation which captures the key elements of the features present, and tries to reconstruct the scene from that understanding).

Gutwin et al. [79] conducted a study measuring the strength of the pop out effects of several different highlighting techniques at 5 levels of intensity each, when placed at different distances (or eccentricities) from the center of the visual field. They presented a field of 104-105 candidate items across 3 monitors arranged in a semicircular arrangement, to cover the participant's entire visual field. They found that motion was a strong visual cue, even when used at low intensities and when positioned far away from the participant's focal point.

3.3.2 Blasting, Flashing, and Tickers – Dynamic Display of Textual Information

McCrickard et al. [118] and Maglio et al. [111] evaluated three different methods for dynamically displaying snippets of textual information:

- **Blasting** – Replacing the text in-place without any transitions,
- **Tickering** – Scrolling the text horizontally across the display area at a fixed rate, resembling displays used in stock trading centres or the headlines/weather information displayed on news broadcasts, and
- **Fading** – The text for each item fades in from the background colour and may/may not be accompanied with a slight scaling-up effect.

They found that tickering is most effective at helping users comprehend the content of the notification, while fading "best facilitates reaction" (i.e. users were most likely to notice the notification) [118]. In a second experiment, they also found that smaller displays were more disruptive (both interrupting and eliciting faster responses), and that slower displays were better for giving time for users to comprehend the content. It is important to note that once again, in this study, there were primarily testing notification techniques which were text-centric, which may explain the "slower-displays for better comprehension" finding.

3.3.3 Effectiveness of Menu Highlights

Several studies have compared the effectiveness of different schemes for making items in menus easier to find. Findlater et al. [64] developed the "Ephemeral Adaptation" technique for optimising item-selection time by sub-consciously priming the user to noticing a subset of items from the menu, by displaying them several milliseconds earlier than other entries. Cockburn et al. [51] developed the "Morphing Menus" technique (i.e. the fontsize of menu items was controlled using a Degree-Of-Interest function [71] based on their relevance/importance), and compared the effectiveness of this technique with more traditional menu designs (e.g. "Split-Menus", where the more relevant items were shwon in a separate section).

3.3.4 Effectiveness of Applied Highlights

Other studies have studied the effectivness of different highlighting techniques in applied settings. Examples of these include Anderson et al.'s [12] study of the suitability of using different types of highlighting techniques for security popup warnings, Davies and Beeharee's [57] studies of different in-game notification techniques, Agapie et al. [10] investigated whether a coloured halo/glow border around text fields may encourage the user to type longer and more descriptive search queries, and Bergman et al. [34] investigated whether graying-out less important files helped the user locate files of interest faster.

4 PCCH – A Framework for Construction and Control of Interactive Highlighting Techniques

To better understand what a highlighting technique is, we present a new structured design framework (PCCH) that systematically describes how highlighting techniques are constructed and how they behave over time.

Prior attempts at creating design frameworks for highlighting techniques [37, 106, 119] have generally followed the partitioning of the design space popularised by the Gestalt Principles [102]. Since this approach is commonly taught in standard visual arts and graphic design textbooks, we shall refer to this approach as the "Standard Graphic Design Model" (SGDM).

In the SGDM's, visual effects are typically divided into several non-overlapping classes: Colour, Contrast, Size, Shape, Texture, Orientation, and Motion. These categories make sense when considering that SGDM models were originally created to help visual artists and designers to develop their intuitions about each effect, and how that could be utilised to convey quantities/relationships and meaning [37]. Also, it should be noted that this taxonomy was created when print-based media (e.g. static displays and printed materials, instead of dynamically updating screens) were dominant, and motion/temporal effects were mostly an afterthought.

The SGDM approach is however less suitable for describing and considering modern computer interfaces. In particular, there are deficiencies for dealing with the types of complicated, dynamic, real-world visual entities such as GUI widgets and the highlighting techniques applied to them that are commonly used. Examples of the mismatches between SGDM's and modern computer interface environments include:

1. Widgets in GUI's are typically described using a hierarchical "Object Orientated" model, where each widget is built out of a series of visual elements whose appearance is controlled using a set of per-element parameters. For example, while SGDM's treat colour, contrast, and texture/pattern effects as being somewhat independent, unrelated, and largely non-interacting concepts, in practice, the end result is that some pixels on an element/surface being shown on screen change colour as a result of some combination of all three.
2. Similarly, instead of only having one visual effect in action, there are usually multiple visual effects being manipulated and balanced at the same time. For example, motion and temporal change are important aspects of many modern GUI's. The behaviour of many different visual effects can be manipulated over time, and in combination with other effects (e.g. the colours of an item can pulse and change while the item itself moves around the screen).

Thus, SGDM's are less suitable for describing highlighting techniques in an unambiguous way that designers or their support tools can easily process, analyse, and use when predicting user performance with an unknown technique. In that sense, the use of SGDM's to describe highlighting techniques hinders the development of a wide range of tools for assisting designers as described by Rosenholtz et al. [140].

We propose that highlighting techniques should instead be described using a model based on how they are constructed and controlled. Our approach is inspired by the QtQuick/QML environment [53] and the standard approach used in most animation systems for content creation systems (e.g. Blender, Maya, and Pixar's Menv System [135]): objects/entities are animated by creating "avars" (i.e. animated-variables [135]), where a parameter of the object is connected to a 2D curve which describes how the value of that parameter's value changes over time. Similar ideas were recently discussed by Kazi et al. [96] with their work on "Motion Amplifiers" – animation effects which take some animation curve as input, and uses that to modify several other parameters of the transform effects applied to the shapes being animated by their system.

This chapter is divided into four parts. The first part (Section 4.1) reviews the contributions of prior design frameworks and the limitations of those frameworks. The second part (Section 4.2) presents an overview of the key concepts necessary for understanding how highlighting techniques can be described using this framework. The third part (Sections 4.3 and 4.4) presents a more detailed discussion of the concepts and techniques that can be used to construct highlighting techniques. The fourth part (Section 4.5) discusses how highlighting techniques presented in the prior literature can be expressed in terms of this framework.

4.1 Prior Design Frameworks

This section reviews a selection of the prior frameworks in the literature for domains related to highlighting. We present this review in several parts: 1) Frameworks that aim to define and partition the design space based on a number of manipulation primitives, 2) Frameworks that focus on describing particular classes of effects, 3) Frameworks best suited to classifying and analysing the effects of a given technique, and 4) Sets of design guidelines available to designers.

4.1.1 General Frameworks for Graphic Design and Highlighting Techniques

There have been several attempts in the literature to develop "grand unified" frameworks which aim to encapsulate all necessary design guidelines and insights related to the use and manipulation of visual effects. This section presents three such frameworks. A key feature of these frameworks is their presentation of a taxonomy of the design space in terms of a small (5-7 item) set of "Visual Variables" (Bertin [37, 45] and Liang and Huang [106]). While the first two (i.e. Koffka's "Gestalt Principles" [102] and Bertin's *Seminology of Graphics* [37]) have a broad focus on graphic design in general, Liang and Huang's framework is explicitly aimed at covering the design space for highlighting techniques (as used in information visualisations to draw attention to items).

Koffka's "Gestalt Principles" are a well-known set of guiding principles for graphic designers, defining how "good" designs can be achieved. Of greatest relevance for highlighting are the concepts related to *similarity* and *proximity*: that is, the relationships between items can be communicate by using similar colours, shapes, sizes, textures, value/contrast, and by adjusting the relative distances between items [159]. Conversely, this implies that a small subset of items can be highlighted (or have pop-out effects applied to them) by manipulating one of these variables/aspects to make the item appear different to the surrounding items.

According to Carpendale [45], Bertin's "Seminology of Graphics" [37] analysed the design space of graphic design techniques for encoding information in visualisations, and developed a broad theoretical foundation that provides advice for designers on how to choose visual designs for best communicating different types of data. While Bertin's framework is comprehensive, it is aimed more at designers producing static infographics as opposed to interactive and dynamic interfaces. In that sense, it has a narrower focus and is more practical/pragmatic than the more abstract/high-level Gestalt Principles. MacEachren [110] extended Bertin's framework to include six new visual variables for motion-based effects: Frequency, Duration, Movement, Rate of Change, Order, and Synchronisation.

Liang and Huang [106] developed a framework for highlighting techniques in information visualisations. Their framework – "*The Elements of Highlighting*" – divides the design space into five *elements* that can be used for *differentiation* of items (see Table 1 in their paper [106]). Each element has several *visual variables* that can be at some discrete level of intensity (e.g. low/medium/high, slow/fast, or small/large). However, the scope of this framework is narrower than Bertin's (i.e. for highlighting techniques as used in information visualisations, instead of graphics in general). Liang and Huang also attempted to address some of the other issues that designers of interactive visualisations need to consider, such as where highlighting fits within interactive architectures, and what it can be used for within a visualisation [106].

4.1.2 Specialised Frameworks for Narrow Classes of Highlighting Techniques

On a more practical level, there have been multiple frameworks devised for particular classes of visual effects and interactive techniques. Unlike the frameworks described in Section 4.1.1, the following examples focus on a particular type of effect (e.g. ways of making animated icons).

Harrison et al. [81] developed a design framework for "Kineticons", or animated effects which only affect the geometry (but not the pixel-space contents) of graphical elements. The Kineticons paper/framework presented two main contributions: 1) A taxonomy defining the difference between "Kineticon" effects and "Animated Graphics" [1] ; and 2) A taxonomy for the origin/source of inspiration for different types of Kineticon effects. Kineticon effects are similar to the "moticons" described by Bartram [29].

Harrison et al. [80] also developed a design framework for controlling the brightness of a LED-based indicator (e.g. the status light on a smartphone) to communicate different states to users.

[1] "Kineticons" modify the geometry (e.g. bouncing dock icons in MacOSX) whereas "Animated Graphics" modify the pixel-space contents only (e.g. the periodic "sheen" that passes across Aero Glass progress bars)

Gluck et al. [74] developed a partial framework based on the types and rates of change involved (e.g. Single State Change, Continuous Slow State Change, Continuous Fast State Change, Continuous Location Change). Similarly, Dragicevic et al. [59] developed a partial framework for types of transitions between two keyframes (e.g. Constant, Slow-In/Slow-Out, Fast-In/Fast-Out, Adaptive Rate).

Anderson et al. [12] describe different ways that "polymorphic warnings" (e.g. dialog boxes displaying security warnings whose appearance changes every time they are used) can use to make users focus on the contents of those warnings instead of dismissing them.

4.1.3 Classification Frameworks

McCrickard et al. [119] developed the *IRC* model for classifying and evaluating notification techniques/systems in terms of three criteria: *Interruption*, *Reaction*, *Comprehension*. The way the IRC framework is presented means that it is most useful as a *classification* system. It also has some limited uses in helping designers identify the *expected outcomes* (i.e. how important is it that users are interrupted, respond immediately, and understand and remember the message being conveyed) that their design must support. However, beyond providing guidance in the initial stages of the design process (and again when reviewing the outcomes of each design), their framework does not address the critical issues of *how* designers should go about achieving the goals required (which the other frameworks discussed here do).

Tomitsch et al. [153] developed a taxonomy for ambient information systems (i.e. minimally disruptive highlighting/notification systems that are designed to operate in peripheral vision). Compared with the IRC framework, Tomitsch et al.'s taxonomy is more detailed, as it defines multiple variables to use when classifying a technique, with each variable able to be defined at 3 levels of intensity/strength. In contrast, with McCrickard et al.'s framework [119], it is often only possible to make True/False assertions about each factor (e.g. it is either important or not important that the user can recall what the purpose of the highlighted message was).

4.1.4 Design Guidelines

Design Guidelines aim to encapsulate the result of usability studies and "best practice" knowledge into a set of rules-of-thumb or guidelines that designers should follow. Several notable examples of these include the W3C guidelines for web developers [1], the "*Hierarchy of Color Usage Guidelines*" from the NASA AMES Research Centre [17, 18], and the guidelines presented in Bartram's thesis [28].

4.1.5 Limitations and Knowledge Gaps of Prior Frameworks

From the preceding sections, it can be seen that there have been a wide variety of prior design frameworks of relevance to the HCI community when working with highlighting techniques. However, there does not appear to be any existing framework that adequately caters to the complexities faced by designers when working with interactive highlighting techniques. In particular, there are 3 key opportunities for addressing the limitations of the existing frameworks:

1. **Broad Scope** – There is a lack of insight into the overall structure the design space. For example: (a) How do different visual effects/manipulations interact with each other? (b) What are the relationships between the specialised frameworks identified in Section 4.1.2 and the "building blocks" identified in Section 4.1.1? It is often unclear that there is a common body of knowledge shared between the InfoVis, Notifications, Security Alerts, Animated Effects, and Graphic Design domains.
2. **Precision** – Most of these frameworks address *what* can be varied, but do not adequately describe how these effects can be controlled in a consistent and precise way. Thus, there needs to be a common vocabulary for precisely describing these manipulations.
3. **Instruction (Design Guidance)** – There are opportunities for helping designers better understand the fully scope and diversity of options that are available. However, some have noted concerns that traditional sets of "Design Guidelines" may not be the most optimal tool for providing this guidance to designers due to the complex multivariate nature of the domain [140].

In the following sections of this chapter, we present a new design framework to address the knowledge gaps posed by the existing frameworks.

4.2 Overview of Design Framework

This section presents an overview of the main components of our design framework. Our framework is developed with the practicalities of implementing and evaluating highlighting techniques in mind, focussing on issues such as:

- **Parametric Control** – How can we describe/specify visual effects *objectively*, in a way that computers (and humans alike) can interpret their descriptions *unambiguously*?
- **Structure** – What is the relationship between all the different types of visual effects? How do they fit together or be applied to UI elements?
- **Implementation Issues** – How can the capabilities of graphics toolkits be used to achieve the effects required?
- **Suitability for Purpose** – How can existing techniques be described in this way? What combinations of techniques may have been overlooked in the prior literature?

4.2.1 Layers of Construction: Constructing Widgets for Highlighting

User interfaces are made up of various UI components or widgets. Each widget is itself composed of a number of visual elements, such as rectangles, images, and text. Figure 4.1 shows a generalised schematic of how a typical widget (e.g. a button, an icon in a grid of items, or even the cursor) can be constructed by building up several layers of visual elements to create a “widget” on screen. Not all of these layers are necessarily visible at all times. For example, the *Underlay* and *Overlay* layers are used to display visual elements (such as a shaded rectangle to indicate that the widget is selected, or a “red number badge” as found on mobile app icons) which are added to the widget when the item is being highlighted, and absent at other times.

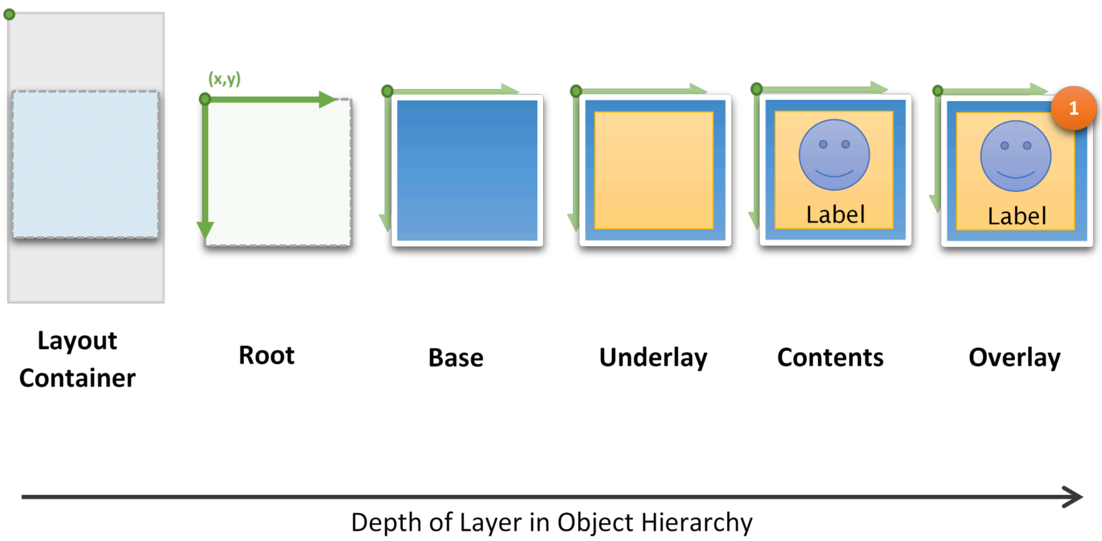


Figure 4.1: *UI Widgets can be constructed by stacking layers of visual elements on top of each other (shown here from deepest to nearest Z-depths)*

The purpose of each element layer in Figure 4.1 is as follows:

- **Layout Container** – This is the layout manager (e.g. a list, box, or grid layout) or the window/drawing canvas that owns the widget. It is not part of the widget.
- **Root** – This is the root of the widget’s hierarchy – all other elements are children or descendants of this. It is is used to position the widget relative to the layout container it is housed in. In “static / in-place” widgets (i.e. those which do have any dynamic transformations applied), the *Root* and *Base* are often the same element to save memory. However, highlighting effects which involve dynamically transforming the elements (e.g. translating, rotating, or scaling them) should have those transforms applied on the *Base* instead, as it is easier to define and reuse those highlighting effects in different contexts without worrying about losing the widget’s original place in the container.
- **Base** – This is the shape that defines the visual boundaries of the item. It defines the background fill colour/texture for the widget, and any borders which may be applied to those. For example, the *Base* layer for a button would contain the border and gradient-fill that define the shape of the button.
- **Underlay** – This optional element is overlaid over the *Base* (but below the *Contents*), and is usually used to show “selection bubbles”. These need to be drawn above the

Base (so that they are visible) but below the *Contents* (so that the contents are not occluded or tinted).

- **Contents** – This layer contains any semantic content/labels which get applied to this widget, such as icons and label text.
- **Overlay** – This optional layer contains any elements which are meant to appear layered above anything else in the widget. A common example of this is the "red dot number badge" (e.g. as shown in Figure 4.1 for the *Overlay* layer).

4.2.2 Parameters of Presentation

Highlighting effects can be created by manipulating the appearance of the visual elements in a widget by applying parametric "Pixel-Level" and "Object-Level" manipulation effects to them (Section 4.2.1). Each of these effects can be controlled using a number of "Parameters", which control aspects such as the strength/magnitude of the effect, the colours used, or the direction of movement. F-Curves (i.e. functions of the form: $v = f(t)$, where v is value the value of a parameter at time t) can be used to animate (i.e. control over time) the behaviour of a parameter.

For example, many Kineticon [81] effects (e.g. making visual elements shake, jump, or bounce), can be created by applying transformations to elements in the *Base* layer (i.e. "Object-Level Manipulations"), and using F-Curves to animate the amount by which the translation/rotation/scale transformations affect the elements. Consider the bouncing/jumping dock icons in Mac OSX: the *Base* layer of the dock-icon widget has a *Translation* effect (an Object-Level effect) applied to it; to create the bouncing motion (i.e. the highlighting effect), the `Y-Offset` parameter of the *Translation* effect is animated by controlling/animating its value using a sinusoidal F-Curve.

Section 4.2.3 discusses how F-Curves can be defined and higher-level control structures for coordinating the use and application of different F-Curves over the life-cycle of a highlighting technique. Some of these higher-level (temporal) control structures also define their own parameters (e.g. `Frequency`, a parameter controlling the speed of animation, defining the number of times per second that an animation, like a single-bounce of the aforementioned Kineticon, gets repeated). We refer to the set of all parameters – from those controlling the per-element manipulation effects (e.g. `Y-Offset`), to those affecting the animation/control structures (e.g. `Frequency`) – as "*HL Parameters*".

It is useful to describe and refer to highlighting techniques in terms of their *HL Parameters* for several reasons:

- **Objective and Unambiguous Description of The Effect** – Referring to each highlighting technique in terms of the parameters used to achieve that effect simplifies the process of replicating prior setups by providing a consistent vocabulary for describing the techniques. Thus, it is easier for designers to build on and directly benefit from prior results. It is also easier for designers to use such parametrised descriptions to translate their ideas into functioning prototypes. This in turn improves the feasibility of computer aided design tools (CADT's): designers can iterate over design ideas quicker (as they only need to input the relevant parameters, instead of building full prototypes), empirical results can be more readily added to shared knowledge bases, and a wider range of approaches (e.g. many popular machine learning approaches) for implementing CADT's become feasible. CADT's may also be able to suggest which parameters could be tweaked to optimise a highlighting technique for particular Noticeability/Distraction targets.

- **HL Parameters help establish the bounds of the design space** – Each parameter has a range of valid/sensible values that can be used. For example, distances (e.g. `Y-Offset`) are bounded by the resolution of the screen (i.e. typically 1920×1080 or less for commonly-used "HD" displays, or between 5-10 thousand on the long-side [169]), rotations (e.g. θ, ω, or ϕ) are bounded to $\pm 360°$, frequency to the 0-8 Hz range (see Section 2.2.4), and colours to 0-255 discrete steps. Even if physical units are used instead of those used by the current-generation display technologies (e.g. using millimetres for distances, or using a percentage-based primary-colour-intensity measure for colour), similar limits would still apply (e.g. screens are still fixed-sized, the human visual field only covers a certain area [159], and the human visual system can only detect a fixed number of discrete colours [17]).

4.2.3 Levels of Control: Creating and Controlling Highlighting Effects

Figure 4.2 provides an overview of the control hierarchy that is used to determine how the widget's parameters are manipulated over time.

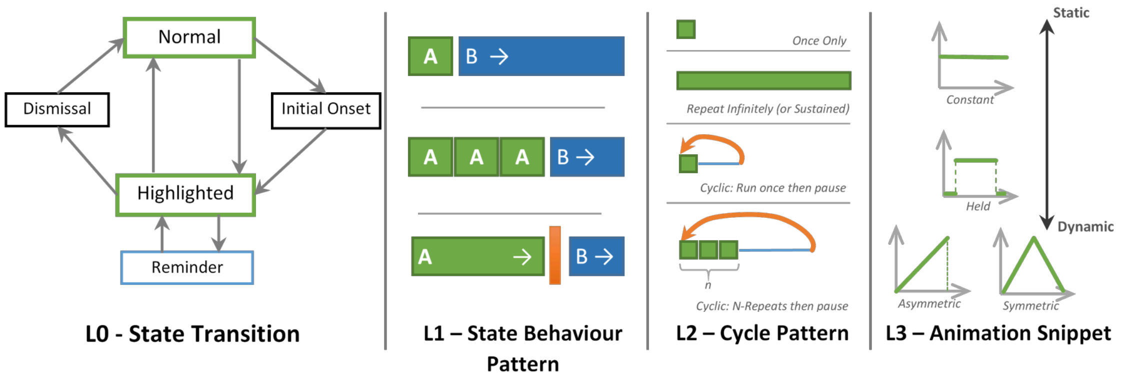


Figure 4.2: *Overview of the different levels at which control over how the highlighting technique behaves is achieved, along with examples of techniques which can be used at each level to do so. The L3 animation snippets control how the parameters of visual elements in the widget change over time*

The purpose of each tier is as follows:

- **State Transition Model** – Highlighting techniques can be in one of several states at a time. The State Transition Model defines what these states are, and how these states relate to each other. Each state has a set of animations and manipulations/effects which will be applied when that state is active. Commonly, highlighting techniques only use the *Normal* (i.e. no highlighting) and *Highlighted* states.

  The other included in this diagram (e.g. *Initial Onset*, *Dismissal*, and *Reminder*) are needed to describe the more sophisticated techniques often found in commonly used interfaces. For example, the Windows Taskbar uses *Initial Onset* to flash an item 3 times, before transitioning to showing a static orange shade for the *Highlighted* state. Another example is the 'starring' feature in Picasa, where the *Normal/Highlighted* states are indicated using the colour of the corresponding toggle button, while the *Initial Onset* and *Dismissal* states are for showing a short animated effect of a large spinning star to alert the user that the state just changed.

- **State Behaviour Patterns** – This defines how a state behaves when it is entered or executed. These patterns describe the relationship between state changes and the way in which Cycle Patterns are played.

  For example, this models whether the cycle is repeated once before immediately entering the next state (e.g. the fading star icon used in Picasa for the *Initial Onset* and *Dismissal* states uses the first L1 pattern in Figure 4.2: `A, B`→ pattern), or whether the current state is maintained until some external event (i.e. a click, keyboard command, or system generated status change such as an email arriving) results in a state change (i.e. third L1 pattern in Figure 4.2: `A`→ `|!|` `B`→).

- **Cycle Patterns** – This pattern controls when an animation snippet is repeated and what gaps or pauses are used between each repetition.

  For example, consider the highlighting technique used to indicate a "suggested move" in Candy Crush Saga: the items being highlighted flash three times, pause for moment, then flash three times and pause again, ad infinitum. This is an example of the "Cyclic: N-Repeats then Pause" Cycle Pattern shown in Figure 4.2.

- **Animation Snippet** – Animation snippets are short clips or snippets of animation which define what "one unit" of animation in the Cycle Pattern does. They contain a set of animation curves (commonly known as "*F-Curves*" in most 3D content creation tools, or "*Avars*" [135]). Each F-Curve describes how a single parameter of an element in the widget behaves over time. More formally, each F-Curve defines the relation, $y = f(t)$, where y is the value of the parameter being animated by the F-Curve, t is the current time (in frames), and $f(x)$ is the F-Curve evaluation function. Common examples of animation snippets include sinusoidal oscillation around the original value, a simple on-off pattern, or physically-inspired effects such as "bouncing" or "jumping".

4.3 Construction of Highlighted Widgets: Element Manipulations

Section 4.2.1 discussed how widgets are constructed from layers of "*Visual Elements*". This section discusses how Visual Elements can be manipulated to create highlighting effects. We argue that Visual Elements are effectively abstractions/generators of "*Pixel Buffers*" (Section 4.3.1), and that the appearance of these pixel buffers can be manipulated to create highlighting effects by applying *Pixel-Level* (Section 4.3.2) and *Object-Level* (Section 4.3.3) effects to them.

4.3.1 Visual Elements are Pixel Buffer Generators

Visual Elements are abstractions used for describing the graphical components that appear as part of a widget. Examples of Visual Elements include various shapes (e.g. Rectangles, Ellipses, Polygons, and Curves), images, text, particle systems, procedural texture generators, and pre-packaged combinations of these that UI toolkit may provide (e.g. any subtype of `Item` in QML [53]). However, this does not yet answer the question of how Visual Elements are related to Pixel Buffers (or what Pixel Buffers are).

The first key insight is that all visible parts of a GUI need to be rendered as a grid of pixels on a screen. A *pixel* is a point sample representing the colour at point (x, y) on the screen. In this thesis, we will refer to a two-dimensional grid/array of pixels as a "*Pixel Buffer*". Common examples of Pixel Buffers include "images" (e.g. photographs, artwork, and screenshots) and the "screen" itself. The screen's Pixel-Buffer is a bit of a special case in that applications draw their GUI's to a special Pixel Buffer known as the "Frame Buffer" [147]; the graphics card then displays the contents of the Frame Buffer to the user using the computer's monitor(s).

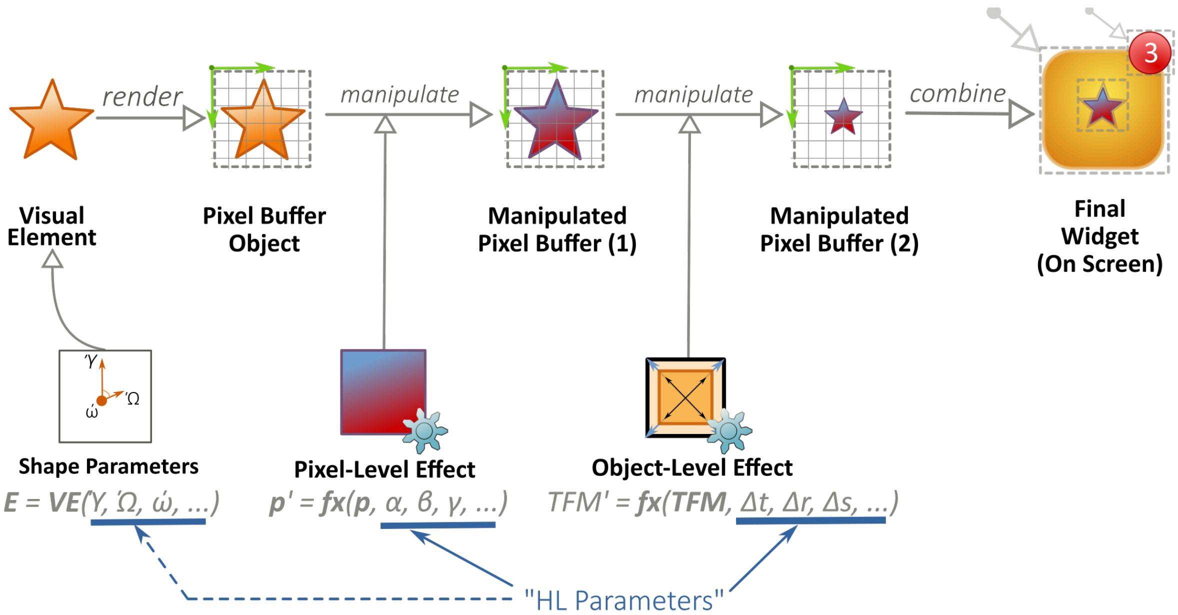


Figure 4.3: *Illustration of the relationships between Visual Elements, Pixel Buffers, Pixel and Object level manipulations, Highlighting Parameters, and UI Widgets. It depicts a typical sequence of operations performed to render a Visual Element on screen as part of a UI Widget. Multiple manipulations/effects can be performed before the Visual Element's Pixel Buffer gets combined (or composited) with those of the other elements in the widget (in the final step).*

The second key insight is that each Visual Element needs to be "rendered" or "rasterised" into a Pixel Buffer so that it can appear as part of the application's UI. To facilitate easier manipulation of the Visual Elements in a widget, each Visual Element gets rendered to its own Pixel Buffer first (i.e. the first step in Figure 4.3), before manipulation effects are applied (i.e. second and third steps in Figure 4.3), and then finally, all the visual elements are combined or "*composited*" together into the Frame Buffer (i.e. final step in Figure 4.3). Thus, Visual Elements are abstractions for generating Pixel Buffers.

As shown in Figure 4.3, Pixel Buffers can be manipulated by applying Pixel-Level and Object-Level effects to them. The effects do not need to be applied in the order shown (i.e. Object-Level effects can be applied before Pixel-Level), and there are no restrictions on how many of each are present. By animating the variables/settings for each effect (i.e. the function arguments underlined in blue), interactive highlighting effects can be created. These variables/settings are the "*Highlight Parameters*".

4.3.2 Pixel Level Manipulations

Pixel Level Manipulations create highlighting effects by changing the *contents* of a Pixel Buffer (by affecting the colours of the individual pixels in that buffer). For example, a highlighting technique where the colour of an icon changes from gray/monochrome to multicoloured (for example, the Copy/Paste icons in Microsoft Word when the Copy/Paste tools can be used) is an example of a Pixel Level Manipulation.

This section examines Pixel-Level manipulations of Pixel Buffers at different levels of abstraction:

- **Low Level** – Section 4.3.2.1 (Colour Models) examines manipulations that are performed on a pixel-to-pixel basis, by changing the colour of each individual pixel.
- **Mid-Level** – Section 4.3.2.2 (Compositing and Blending) presents techniques for controlling how Pixel Buffers layers are combined/merged together.
- **High-Level** – Section 4.3.2.3 (Per Element Colours) provides a high-level perspective of how Pixel-Level effects can be applied in terms of the semantic components of the Pixel Buffer (for example, how the outline or interior shading of the element the buffer represents will look).

4.3.2.1 Colour Models

At the most elemental level, highlighting effects can be achieved by manipulating the colour of individual pixels of a Pixel Buffer. To understand how these manipulations can be performed, it is important to first understand how colour is represented on computers.

Colour can be represented in many different ways. The accurate representation and reproduction of colour is a complex topic that is not well understood by many people. Notably, many people will only have been exposed to RGB tuples and HSV/HSL from their use of common office and graphics packages (e.g. Paint.NET). However, graphics professionals need to have a much deeper understanding of these issues and how these can affect the way that their work is seen when deployed (e.g. the problems that differences in monitor colour accuracy and calibration pose when doing colour-critical work). Caution is advised before delving down the "rabbit hole" of colour science, as a considerable time investment is required to gain a complete understanding of these issues.

This section highlights four of the most useful and important colour models for working with highlighting techniques: RGB, HSV or HSL, the L*ab colour space, and the XYZ colour space. The RGB and HSV/HSL representations are optimised for ease of use and computational efficiency, at the expense of perceptual fidelity and accuracy. In contrast, the L*ab/CIE and XYZ colour spaces favour perceptual fidelity/accuracy at the expense of being less intuitive and more computationally intensive to use.

4.3.2.1.1 RGB – Red, Green, Blue

Colour is often represented as an additive combination of red, green, and blue light (RGB), where the intensities (specified as a percentage of "full strength") of these three primary colours are adjusted to produce different colours [159, 166].

Many different "RGB colour spaces" can be defined by choosing different reference colours for the 3 primary colours. The two most popular standards-based RGB colour-spaces are *sRGB* ("Screen RGB") [13] and *Adobe RGB* [150]. Of the two, sRGB is the most common and popular [166]; most hardware devices (e.g. monitors and cameras) and most software (with the exception of specialist tools) use this as either their sole/primary colour representation. However, sRGB is limited in that it is not capable of representing all the colours that the human eye can perceive [159], particularly in terms of certain shades of red and green [150].

4.3.2.1.2 HSV – Hue, Saturation, Value

The *HSV* (Hue, Saturation, Value) or *HSL* (Hue, Saturation, Lightness) colour models were designed to allow easy manipulation and selection of colours in a computationally efficient manner, in a way that is more similar to our intuitive descriptions of colours. Therefore, HSV/HSL are the most convenient colour models when controlling and constructing HL's.

Figure 4.4 shows that this colour model uses the following 3 variables:

- **Hue** – This represents the "base colour" that is used. It can be viewed as a selector for the wavelength of light from the visible light spectrum, expressed as a factor from 0 to 1 (0 = Red, 0.5 = Cyan, 1 = Pink).
- **Saturation** – This represents how vibrant or "not grey" the colour is. It is often manipulated by highlighting techniques to communicate different levels/states of some variable. For example, many UI's use *Desaturated Colours* (i.e. shades of grey, $S = 0.0$) for the "Disabled" (*non-highlighted*) state of a tool's icon, and *Saturated/Colourful Colours* (i.e fully saturated, $S = 1.0$) for the "Enabled" (*Highlighted*) state.

  Saturation is often used as a computationally efficient approximation for a "colourfulness" control. Recent studies such as [82, 138] have developed simple models for predicting "colourfulness". However, to date, these are more useful for post-hoc analysis and evaluation than for active manipulation, as the current formulations are only for computing a colourfulness metric from some given RGB colour.
- **Value or Lightness** – This represents whether the colour tends toward black or white (or alternatively, darkness and lightness).

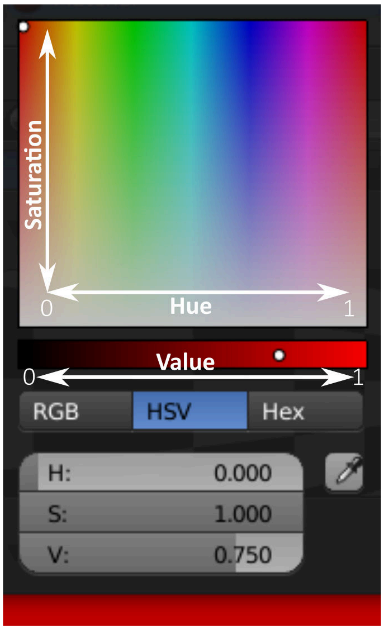


Figure 4.4: *Screenshot of a HSV colour picker/selector, with annotations showing the effects of the Hue, Saturation, and Value parameters.*

4.3.2.1.3 CIELab – Perceptual Colour Space

The CIELab colour space (`Luminance, a, b`) colour space is an alternative way of representing colours [24]. Unlike the HSV model, it is an example of a "perceptual" model. Perceptual models aim to ensure that colours which appear "equally bright" to human observers have the same luminance value [159]. In theory, this makes it easy to perform colour manipulations to obtain colours which will not suddenly appear overly dark or bright when placed alongside others with similar values, making this a useful property for ensuring that colour changes should not unintentionally create highlighting effects. Conversely, it should hold that if we deliberately manipulate the luminance levels, we should be able to obtain two different colours which are perceived as having different perceptual brightness, thus achieving highlighting effects.

However, in practice, colour models are not perfect representations of real human colour perception. Instead, they are based on a series of mathematical approximations which transform the sRGB model into a colour space which closely resembles the shape implied by the measured samples. This is evidenced by the numerous versions and revisions of the ΔE (delta-E) equations (i.e. CIE76, CIE94, CIEDE2000) for computing the "difference" between two colours using Lab colour space, where the subsequent revisions were made to "correct for biases and inaccuracies" [165]. For example, the CIE94 standard was created to account for issues where certain colours (notably blues and yellows) were being unfairly treated [165].

4.3.2.1.4 XYZ – Reference Colour Space

Another colour space which should be briefly mentioned is the XYZ colour space. This is known as the "reference" colour space containing every possible colour – including many of which are not able to be represented by any monitors [159] (p. 113). It is of lesser practical value when dealing with highlighting technique than the other models presented here, as it is relatively difficult to intuitively understand the nature of this abstract 3D colour space [159]. However, it should be noted that this space is often used as an intermediary when dealing with specialist applications where colour accuracy when performing manipulations is essential.

4.3.2.2 Compositing and Colour Blending

Compositing refers to the process and techniques for combining several layers of pixels together into a single pixel buffer (i.e. the "*accumulation buffer*"). For instance, when rendering the layers of Visual Elements in a widget, there are situations where parts of different Visual Elements overlap each other. To resolve these conflicts, "Z-Depth" values are used to determine the order in which elements are drawn into the accumulation buffer using *Painter's Algorithm* [70]: Elements with the largest Z-Depth values are drawn first, and those with the smallest Z-Depth values are drawn last (i.e. from back to front).

Colour Blending is used to refer to the technique used during the compositing process to determine what colour should be used for a particular pixel. When adding an element into the accumulation layer, the opacity (or "alpha") values for each pixel are used to determine the amount that the colour of that pixel contributes to the colour stored in the corresponding pixels in the accumulation buffer. Lower opacity values mean that the elements's colours contribute less to final colour, while higher opacity values mean that the elements's colours contribute more. Note also that like RGB colours, opacity can vary per pixel: so parts of a layer may contribute more than others.

From these concepts, there are two controls (or parameters) that can be manipulated by highlighting techniques to control the appearance of the widget: Opacity and Z-Depth.

- *Opacity* controls the amount that the element contributes to the accumulation buffer, and
- *Z-Depth* controls the order in which elements are rendered.

Elements with high opacity and low Z-Depth can occlude those with higher/deeper Z-Depth values. Together, these two controls are most useful for controlling the visibility of elements (complete with transitions between the visible and invisible states).

4.3.2.3 Per-Element Colours – Objects and Textures

At the highest level, there is also the issue of what colours are assigned to which parts of elements. In the majority of graphics toolkits, each element can be split into two regions: Fill and Border.

The *Fill* is the pixel region which occupies the "body" or interior of each element, while the *Border* is the pixel region which occurs along the edges of each element surrounding the Fill region. For example, in the *Base* shape in Figure 4.1, the *Fill* corresponds with the blue shaded region, while the *Border* is the white shaded region around the *Fill*. The *Fill* and *Border* regions can be manipulated independently of each other, meaning that different colours can be applied to each of these regions. In addition to manipulating the colours of these, it is also possible to manipulate the proportion of the shape that the border occupies relative to the fill.

The appearance of each of thee *Fill* and *Border* regions can be controlled by applying colours in one of the following ways:

- **Using a Single Colour** – This technique means that a single colour is applied to all pixels within the affected regions.
- **Setting the Opacity to Zero** – The effect of this is that element has no contributions to the final colour for the pixels in this region.
- **Using a Gradient** – Instead of applying a single colour/opacity value, the colour/opacity for each pixel is determined as a function of its position in the object (relative to the reference points for the texture). Note that this technique can be computationally expensive if dynamically calculated, so in most cases, this is implemented by rendering pre-computed image textures instead.
- **Using an Image/Texture** – Each pixel in the region is mapped to a corresponding pixel or pixel range in a reference image (i.e. "a texture") using texture coordinates [147]. This mapping thus makes it possible to apply detailed patterns, to use intricate shapes/glyphs which otherwise be difficult to define, and to apply other effects to elements without needing to generate extra elements for those.

4.3.3 Object Level Manipulations

Object-Level Manipulations create highlighting effects by applying a controlled geometric manipulation (i.e. transformation or deformation) to the plane/rectangle that the coloured pixels occupy. These manipulations can be characterised in terms of two criteria: type of manipulation, and whether this distortion occurs in 2D or 3D space.

4.3.3.1 Dimensionality of Manipulations – Two or Three Dimensions

Manipulations can appear to be performed in a two dimensional (2D) coordinate space or three dimensional (3D) coordinate space. 2D manipulation techniques are relatively easier to implement and use with the current generation of UI/graphics frameworks. Dedicated support for performing 2D manipulations is often provided by toolkits, allowing developers to easily add such effects without much additional effort.

Implementing 3D manipulation effects however is usually much more involved. For simple transforms, it may be possible to approximate these effects by using a 2D distortion effect. However, in the other cases where this is not possible, more involved methods are needed (such as using various pre-computed sprites and/or special OpenGL contexts). It should be noted that all 2D manipulation techniques can also be performed in 3D. This is because all the 2D operations are performed on a plane (i.e. the X-Y plane). As a result, these transforms can be performed by setting/treating all the Z-components of any vectors involved to zero.

4.3.3.2 Transformation Techniques

Translation, Rotation, and Scaling are examples of affine transformations (i.e. transforms which preserve the distance ratios between points along straight-line). They are also widely considered to be the three primary transformations [145]. For instance, in 3D content creation applications (e.g. Blender and Maya), these are the only transformation tools which are exposed as in-viewport manipulator widgets to facilitate easier direct manipulation of geometry in the viewport.

Highlight Parameters for these transformation techniques are shown as text labels in the diagrams presented in this section.

4.3.3.2.1 Translation

Translation is the simplest of the manipulations. In two dimensions (see Figure 4.5), it is equivalent to redrawing/offsetting a block of pixels horizontally, vertically, or diagonally (i.e. along both horizontal and vertical axes at the same time) by a certain amount (α) which we refer to as the *amplitude* of the motion.

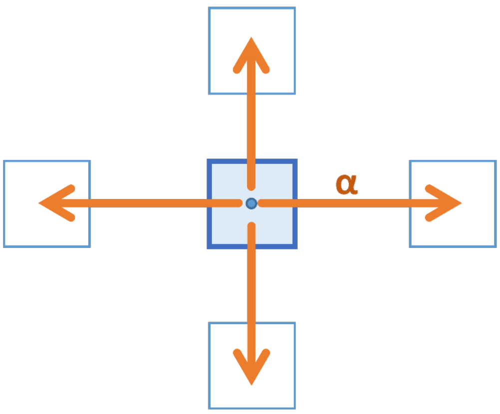


Figure 4.5: *Translation in two dimensions. The element can be translated along the x (horizontal), y (vertical), or both (not shown), by amount* α

For three dimensional transformations, in addition to the two-dimensional behaviours, it is possible to have movements on the third dimension (i.e. depth – or forward and back along an axis perpendicular to the screen/view plane). In many cases, depth-axis translation can be simulated by scaling the object.

4.3.3.2.2 Rotation – 2D

In two dimensions, rotations are performed by using the midpoint of the element as the pivot point, and rotating the element around the screen's normal axis (Figure 4.6). This axis can be conceptualised by imagining a pole protruding from the dot representing midpoint, and spinning the element around this plane as if it were skewered by that stick. Rotations around this axis can occur in either clockwise or anti-clockwise directions, depending on whether the *amplitude* (α – i.e. the parameter controlling how far the item is spun around the axis) is positive or negative. An example of this technique is the "Rumble" technique [81] which is used on iOS when editing app icons.

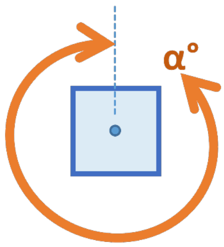


Figure 4.6: *Rotation in two dimensions. The element is pivoted around its midpoint by α degrees*

A related technique is to have the pivot-point placed outside the element (see Figure 4.7). The result of this is that the element appears to move around in a circle instead of wobbling/pivoting in place.

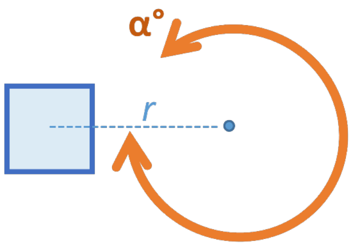


Figure 4.7: *An alternative way of rotating an element in two dimensions. Here the pivot point is placed outside the element. There are three parameters here: the **rotation amplitude**, α, the **offset radius**, r, and **pivot tracking***

This technique is of course more complicated than the former, as there are two other parameters which can be controlled (in addition to the amplitude, which still controls how many revolutions/cycles the element has been rotated). The first is the *radius* (or the distance from the pivot point to the midpoint of the element), and the second is *pivot tracking* (or whether the item is rotated so that the same side always faces the pivot point). By manipulating these two additional parameters, a variety of different effects can be achieved. For example, the radius can be manipulated to make the object trace out a spiral (i.e. by continually shrinking or increasing the radius) or an ellipse (i.e. widening when near the horizontal axis, and shortening when near the vertical), instead of simply tracing out a circle. The effect of the second parameter (i.e. whether the same side continually faces the pivot point) can be used to achieve effects like the Oscillating Blue Arrow found in Mac OSX (for indicating menu items that users are looking for), where the orientation of the arrow stays fixed while it moves in a circle beside the menu item.

4.3.3.2.3 Rotation – 3D

Rotations can be extended into three dimensional space as well, by pivoting the plane around the x (horizontal) axis, y (vertical) axis, or around a pivot point placed in the midpoint or off to the side.

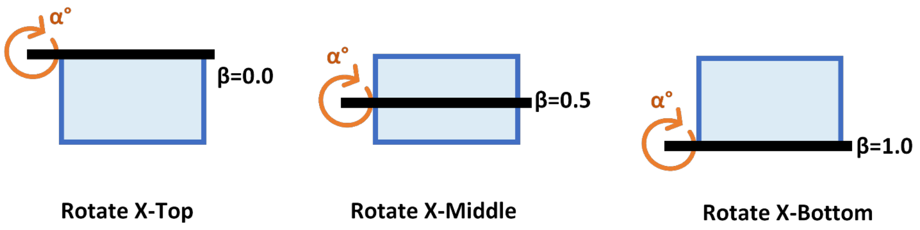


Figure 4.8: *Rotations around the horizontal axis. It is possible to place the rotation axis at different heights (y-coordinates), but these particular configurations are more likely candidates.*

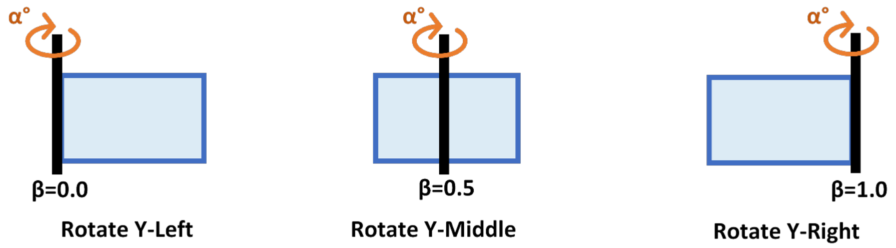


Figure 4.9: *Rotations around the vertical axis*

Figure 4.8 shows examples of pivoting the element plane around the horizontal/x-axis, and Figure 4.9 shows rotations around the vertical-y-axis. In both of these cases, the effect is equivalent to a flag blowing in the wind or flip-flopping around a flag pole. It should be noted that in both cases, it is possible to place the rotation axis anywhere along the perpendicular axis (i.e. Y-axis for Figure 4.8, and X-axis for Figure 4.9). However, doing so often does not make much sense, especially when the elements are small, as the difference between the cases often amounts to 1-2 pixels. Thus, only these 6 examples are shown here.

It is also possible to pivot around a single point instead of an axis. In those cases, the effect is equivalent to mounting a plane to an invisible ball and rotating it. Quaternions or Axis Angle rotation representations (converted to a 4×4 matrix) are needed in that case for controlling the effect to avoid Gimble Lock issues.

4.3.3.2.4 Scale

As Visual Elements are mostly defined as 2D planes, the effect of scaling on these is largely a two dimensional effect.

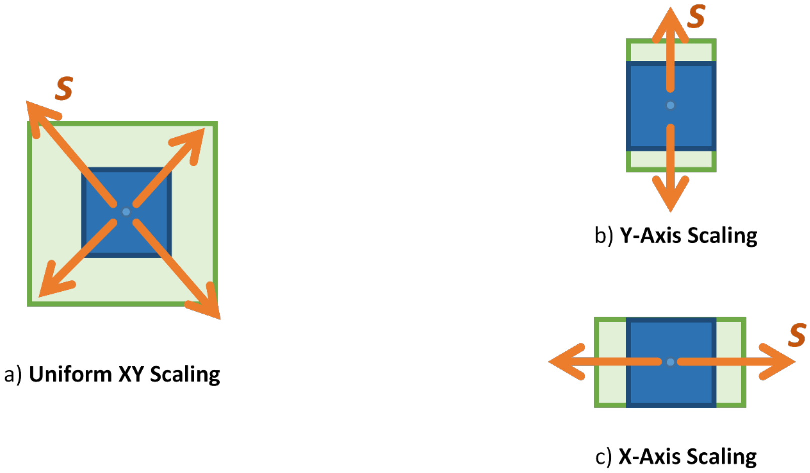


Figure 4.10: *Scale transforms in two dimensions. a) Uniform XY Scaling, b) Y-Axis Scaling, c) X-Axis Scaling*

Figure 4.10 shows three examples of the ways that scaling can be applied to an element. Here, "s" is the scale factor parameter which can be controlled using animation providers. Since this is a *factor*, its effect is multiplicative instead of additive. As a result, the rest/default value here is 1.0 instead of 0.0 for other parameters.

In the *Uniform XY* case, the same scale factor is applied to both axes at the same time. In the two other cases, the scale factor is only applied to one of the axes (vertical and horizontal respectively). It is possible to combine the per-axis effects on the same element to get a non-uniform scale effect.

4.3.3.3 Distortion Techniques

Distortion techniques however work by changing the location of some points within original plane relative to their neighbours. That is, distortion techniques work on parts of the element (and with different effect strengths) instead of the whole element at the same time by the same amount. Distortion effects can be achieved using a deformer, or by directly manipulating pixels painted on that plane.

Deformer-based distortions are achieved by subdividing the visual element into a dense grid-like mesh, and using a secondary control structure (the "deformer") to define the "goal" or target state for the deformed element to assume. The deformer will then adjust the positions of the mesh vertices until the deformed object assumes the target state. There are two main types of deformers which are commonly used:

- **Cage Deform** – Cage, Lattice, or Free-Form Deformers (FFD) are a technique where a grid-like cage-mesh is wrapped around the geometry being deformed. The geometry is deformed by displacing each vertex by an amount proportional to its distance from the nearest cage-vertex (determined from the deformer's "rest position"). While this technique allows for smooth and flexible distortions of the geometry (e.g. for simple "squash and stretch" effects), the regular grid-like cage of control points can be overly restrictive and tedious to control.

- **Skeletal Deform** – Skeletal deformers can be used for more localised and specialised control over the deformations. They are well suited to animating internally-articulated objects (e.g. chains of objects, or for animating humanoid movements). Unlike cage deformer controls, each control point (or "bone") possesses multiple degrees of freedom (i.e. bones can be rotated and scaled in addition to being repositioned). Bones can also be placed in arbitrary arrangements as/where needed. When constructing the widget, the deformed-geometry must first be associated with the skeleton by assigning deformation weights to each vertex indicating the influence that each bone's transform exerts on that point.

Cage Deform is generally used for all "simple" manipulations which affect the whole shape, while Skeletal Deform is used to achieve more targeted effects such as applying a walkcycle to the element [81, 130].

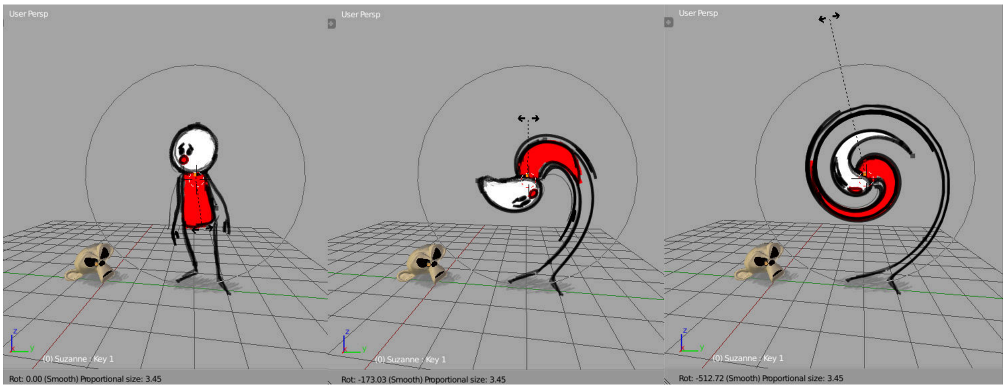


Figure 4.11: *Example of a rotation transform being performed on a pixel-by-pixel basis, with a soft falloff function to include nearby points in the deformation as well.*

Distortion effects can also be created by performing per-pixel operations using a Degree of Interest (DOI) function [71]. The most well known distortion effects using DOI functions are those using proximity-based falloff curves (or Radial Basis Functions). For example, Furnas's "Fisheye Lenses" [71] are DOI distortion effects where points closer to the cursor are magnified more than those further away. Another example of proximity-based DOI distortion effects are "soft brush" tools/deformation effects (as shown in Figure 4.11), where points from away from the selected points are affected less.

4.4 Control over Time: Pattern Languages for Describing Interactive Behaviour

The previous section described how Visual Elements could be manipulated to form highlighting effects. This section addresses the problem of how to model highlighting techniques with temporal effects (for example, highlighting effects featuring effects such as motion, flashing/flickering, colour changes, or whose behaviour depends on user input). In this section, we present the pattern languages and control structures (as listed in Section 4.2.3) used for controlling the behaviour of IHT's.

4.4.1 State Transition Model

An Interactive Highlighting Technique can be in one of several states. In addition to considering how a highlighting technique looks and behaves when it is "active", we also need to consider how it looks and behaves when it is *not enabled*, as well as how it *transitions* between those states. A *State Transition Model* can be used to describe the lifecycle of an IHT as it transitions between different states from the time it first appears until it is dismissed (or cancelled). This lifecycle is shown by the flowchart in Figure 4.12. Each of the boxes represents a state in the lifecycle, while ellipses represent state transitions, and diamonds represent events. Each state in an IHT may use different visual effects. Not all states depicted may actually exist for a given IHT.

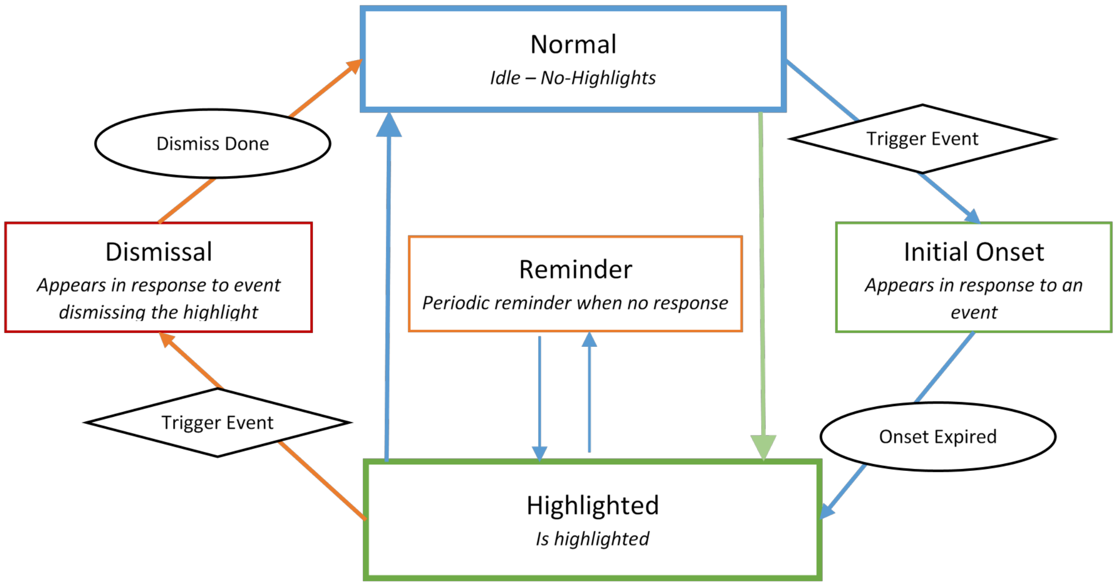


Figure 4.12: *Flow chart of the different states/stages that make up a single highlighting technique. Each state can have a different highlighting effect applied (as necessary). Apart from* Normal *and* Highlighted*, all other states are optional. Boxes represent states, ellipses represent transitions, and diamonds represent events.*

The most basic IHT's require two states: *Normal* and *Highlighted*.

- **Normal** – This is the state of the item as it exists normally, without any highlighting applied. The lifecycle of all highlights begin and end here. For instance, this is the state that applies to all visible elements in a user interface before any highlights of any sort are applied, and is the state that an item returns to after its highlight is removed.
- **Highlighted** – For "simple" highlighting techniques such as those we classify as being "Static" (e.g. a shaded background, border, or a "red dot number badge"), this state is self-explanatory, and represents the state where the additional graphical elements providing the highlighting are visible.

More sophisticated IHT's may require additional states:

- **Initial Onset** – This state is usually used to describe the use of dynamic techniques to notify the user that some time-critical event has just occurred.

For example, consider the Firefox "Downloads Complete Arrow": it shows an animated graphic of the "downloads" icon (with blue shading) growing larger and fading (i.e. *Initial Onset* stage) overlaid over the standard blue-shaded "downloads" icon in the toolbar (i.e. *Highlighted* stage). The *Normal* stage of this IHT shows a grayscale icon instead).

Another example of *Initial Onset* is how Microsoft Windows flashes the taskbar items and titlebars of certain windows three times (i.e. *Initial Onset*) before the orange highlight is held steady (i.e. *Highlighted* stage) if the user did not react. Dynamic effects in such cases are intended to act as a "user interrupt signal", to draw users back to the original task that they were waiting for.

- **Reminder** – This state is used to periodically remind the user that they have not attended to a particular highlighted item. An example of this state is the way that the "Recommended Next Move" is indicated in Candy Crush Saga: the pieces being recommended are pulsed three times (at a rate of 2-3 Hz) followed by a pause of similar duration, before the cycle repeats. As with *Initial Onset*, dynamic effects are usually used in this case as it is believed that the static highlights are being ignored by users. FIgure 4.16
- **Dismissal** – Similar to *Initial Onset*, it is sometimes necessary to use a dynamic highlighting technique in situations where an item transitions from a highlighted state to a non-highlighted state. This is specifically true when such transitions may indicate a safety critical state change (e.g. pilots would want to know that the autopilot was about to disconnect).

4.4.2 State Behaviour Pattern

State Behaviour Patterns (SBP's) describe the behaviour of an IHT when in a particular state. SBP's achieve this by describing the relationship between *HL's / Cycle Patterns* (Section 4.4.3), User Input / System-Status Changes, and when each state should transition to other states. This section introduces the graphical notation we developed to express these relationships in an easily readable format.

Figure 4.13 shows an example of a commonly used SBP known as the "*Hold Until Cancelled*" pattern. The first block (labelled **(1)**) shows that the HL / Cycle Pattern for State '**A**' should be used until some event interrupts it (e.g. the user clicks on the highlighted item; this is denoted using the block labelled **(2)**). The third and final block (labelled **(3)**) indicates that once the event/interruption has occurred, the IHT should move to State '**B**'.

A common use case for the "Hold Until Cancelled" SBP is for implementing the appearance of toggle buttons. For example:

- In the *Normal* state – **A1** is the "Off" icon, and **B** is the *Highlighted* state.
- In the *Highlighted* state – **A1** is the "On" icon, and **B** is the *Normal* state.

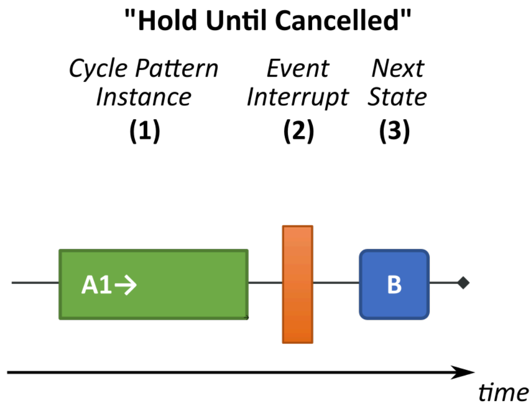


Figure 4.13: *Example of the simple "Hold Until Cancelled" State Behaviour Pattern*

This means that (when starting from the *Normal* state), the IHT will show the "Off" icon until the user clicks on the button. When the button is clicked, the IHT will start displaying the "On" icon instead, as it has transitioned to the *Highlighted* state. When the user clicks on the "On" icon, the IHT will go back to the *Normal* state, and will resume showing the "Off" icon.

4.4.2.1 SBP Notation

Figure 4.14 shows an overview of the different types of graphical notation used for describing SBP's. The figure is divided into three sections:

- The *Top Section* ("*Block Types*") shows the 5 types of blocks that can used in SBP's.
- The *Middle Section* ("*Control Flow*") shows examples of different ways the blocks can be arranged to describe different control flow constructs. The light-grey blocks with dotted edges are placeholders which indicate where blocks should be placed in actual usage.
- The *Bottom Section* ("*Decor*") shows the decorative timeline/axes marker included at the bottom of every SBP diagram to illustrate how to interpret the diagram.

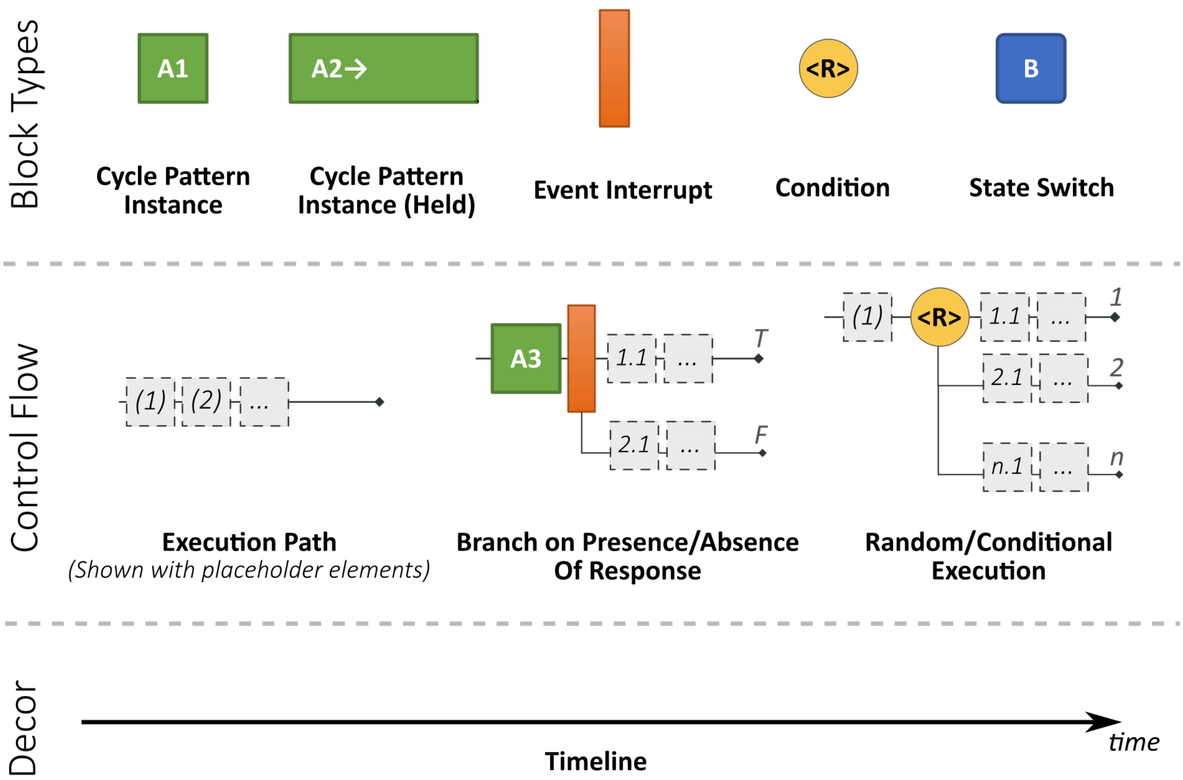


Figure 4.14: *Illustration of the element types used in the graphical notation for SBP's.*

Control Flow in SBP's

SBP can be interpreted by scanning along each line of blocks from left to right, executing each block encountered, and then moving to the next block when the current block terminates. Execution of a SBP ends when a *State Switch* block is encountered.

When multiple execution paths are stacked on top of each other (e.g. in the "Branch on Response" or "Random/Conditional Execution" constructs in Figure 4.4.2.1), these secondary paths represent the alternative paths that may be used depending on the outcome of the *Event-Interrupt* or *Condition* that triggered the forked paths.

Block/State Names

For generality in many of the examples presented here, the CPI and SS block names use `A` to refer to the current state, and `B`, `C`, . . . to refer to adjacent states directly reachable from the current state. However, when actually defining SBP's as part of an IHT, it is clearer to use the following set of single-letter codes to explicitly identify the states used. These codes are:

- `N` = Normal
- `H` = Highlighted
- `I` = Initial Onset

- `D` = Dismissal
- `R` = Reminder

For example, "`H1`" would refer to the first cycle pattern in the *Highlighted* state, "`D2`" would refer to the second cycle pattern in the *Dismissal* state, and "`I`" would be used on a State Switch to the *Initial Onset* state.

4.4.2.2 Block Types

As shown in Figure 4.14, SBP's can contain the following types of blocks:

- **Cycle Pattern Instance** (CPI) – CPI blocks are used to play one of the cycle patterns attached to the current state. Usually, there is only a single cycle pattern assigned to the state, but multiple patterns may present to allow the HL to randomly choose which is shown (e.g. for use with "Random/Conditional Execution"). The number after the '`A`' indicates which state pattern is used (e.g. `A1` plays the first cycle pattern, while `A2` plays the second cycle pattern).

  There are two types of CPI's:

  – ***Normal Duration*** CPI's (e.g. the block labelled "`A1`") – The cycle pattern is played a single time (for however long that takes), and then execution proceeds to the next block.

  – ***Repeated/Held Infinitely*** CPI's (e.g. the block labelled "`A2->`") – This type of CPI is used with an "*Event Interrupt*" block. With this type of CPI, the referenced cycle pattern is repeated/held infinitely many times until the event/signal the Event-Interrupt block was waiting for occurs.

- **Event Interrupt** (EI) – Event-Interrupt (EI) blocks are used to represent a point where the highlighting technique requires some trigger-event/signal to occur before the next block can be executed. There are two types of triggers:

  1. **User Initiated** – The user performed some action such as performing a mouse click, key press, or touch/swipe gesture.
  2. **System Initiated** – A semi-autonomous system process generated a signal that some change/event occurred (e.g. a new message arrived, wifi disconnected, video processing completed).

  Event-Interrupt blocks are not handled in the same way as other block types:

  – When the EI-block occurs after a *Repeated/Held CPI*: Execution can only proceed to the block following the EI when a suitable event (e.g. click, tap, swipe, keypress, or status change) occurs. Otherwise, the CPI keeps repeating until a suitable event occurs.

  - When the EI-block occurs as part of a "Branch on Presence/Absence of Response" construct (as shown in the second SBP diagram on row 2 of Figure 4.4.2.1): Execution depends on whether a suitable event occurred before the *Normal Duration CPI* (i.e. `A3` in the diagram) finished running. If a suitable event occurs, the `True` branch is taken (i.e. the block at position `1.1` is executed). Otherwise, the `False` branch is taken instead (i.e. the block at position `2.1` is executed).

- **State Switch** (SS) – State Switch blocks are used to indicate that the current state has finished, and what state the IHT to transition to.

- **Random/Conditional Execution** (RCE) – Random/Conditional Execution blocks are used to implement IHT's where the HL type used varies each time the SBP is executed. For example, an RCE block could be used when implementing "*polymorphic warnings*" [12] (i.e. instead of always showing warning messages using the same HL technique each time, a pseudo-randomly chosen technique from a predefined set of options is used).

  When an RCE block is encountered, a random number r is generated, such that $1 \leq r \leq n$ (where n is the number of branches/HL techniques that can be used). r is then used to choose which branch is used (see the "Random/Conditional Execution" construct in Figure 4.14, the third diagram on row 2). For example, if r is 2, then the branch/Execution Path starting with the block at position `2.1` will be used.

4.4.2.3 Examples of SBP's

In this section, we present some common examples of SBP's used in highlighting techniques. Many other types of SBP's could be constructed but do not appear to be used in any commonly used highlighting techniques; it is not clear whether those other patterns are not used as they are not useful, or whether designers have not considered those possibilities due to the lack of a design framework.

Here are some examples of representative SBP's:

- **Hold Until Cancelled** – This was shown earlier in Figure 4.13 as the example used to introduce SBP's. It is most often used for the *Normal* and *Highlighted* states.

- **Run Once Then Change** – The Run-Once-Then-Change pattern (Figure 4.15) is most often used for the *Initial Onset* and *Dismissal* states (i.e. states used to transition between *Normal* and *Highlighted*). This SBP is used to play a short one-off animation and then change to the next state (e.g. "*Initial Onset* $\rightarrow$ *Highlighted*" and "*Dismissal* $\rightarrow$ *Normal*").

  The Picasa "big fading icons" star effects (used for the *Initial Onset* and *Dismissal* states on the "Image Starring" IHT) is an example of the Run-Once-Then-Change SBP in action. Upon entering the states where it this SBP is used, an animation runs, showing the icon-overlay (a large star) appearing, rising up to the middle of the screen, and falling back down and disappearing. This animation is repeated only a single time before the IHT enters the *Highlighted* state (i.e. the icon of the toggle button is coloured instead of being a grayscale).

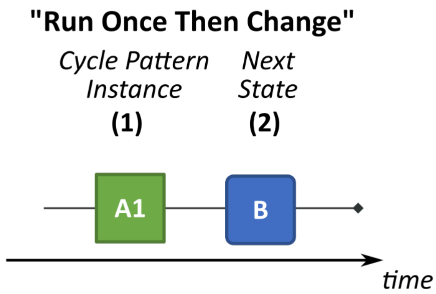


Figure 4.15: Example of the simple "Run Once Then Change" State Behaviour Pattern

- **Interruptible Run Once Then Change** – This SBP is very similar to the previous one, except that instead of forcing the user to wait for the animation to complete before the state changes, a different state change can occur if they respond earlier. For example, this SBP is often used when the IHT has a *Reminder* state (as shown in Figure 4.16): if the user responds to the IHT before the cycle pattern completes, the IHT goes to the *Normal* state (i.e. no further highlighting is needed, as the notification has been attended to); otherwise, it will continue to cycle between the *Highlighted* and *Reminder* states.

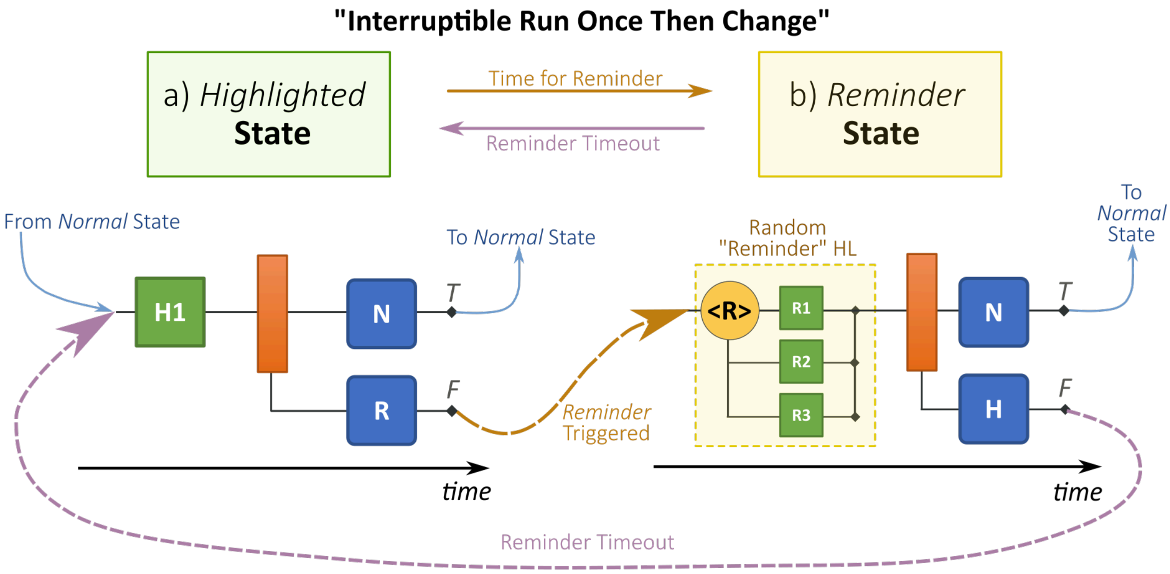


Figure 4.16: Example of how the Highlighted *(left-side) and* Reminder *(right-side) states interact with each other. Both states use the "Interruptible Run Once Then Change" State Behaviour Pattern. Dashed-line arrows show how execution flows from one state's SBP to the other state's SBP. Also note that in place of a single "`R1`" block, the SBP for the* Reminder *state uses a "Random/Conditional Execution" construct (i.e. the yellow shaded box) to randomly choose between one of 3 HL techniques (i.e. `R1`, `R2`, `R3`) for reminding the user that they have not responded yet.*

- **Random/Conditional Execution** – The Random/Conditional Execution pattern can be used to implement "polymorphic warnings" [12]. An example RCE being used in a highlighting technique is shown in Figure 4.16 (within the yellow-dotted box labelled 'Random "Reminder" HL'). When the *Reminder* SBP is executed, one of 3 HL techniques for that state is randomly chosen and used. For example, `R1` could be to flash the icon 3 times with a yellow colour, `R2` could be to flash the icon 4 times with an orange colour, and `R3` could be to flash and shake the icon 5 times with a red colour. Thus, whenever the *Reminder* state is runs, the user would be shown either a yellow flashing icon, an orange flashing icon, or a red flashing icon, before the IHT returns back to the *Highlighted* state.

- **Multi-Stage Interactive Pattern** – The Multi-Stage Interactive Pattern is a novel SBP we identified using this design framework. To our knowledge, it has not actually been described/used elsewhere before. As shown in Figure 4.17, It is formed by chaining a sequence of "Hold Until Cancelled" patterns together within a single SBP (i.e. it contains multiple *Held CPI - EI pairs* in succession before a state change finally occurs).

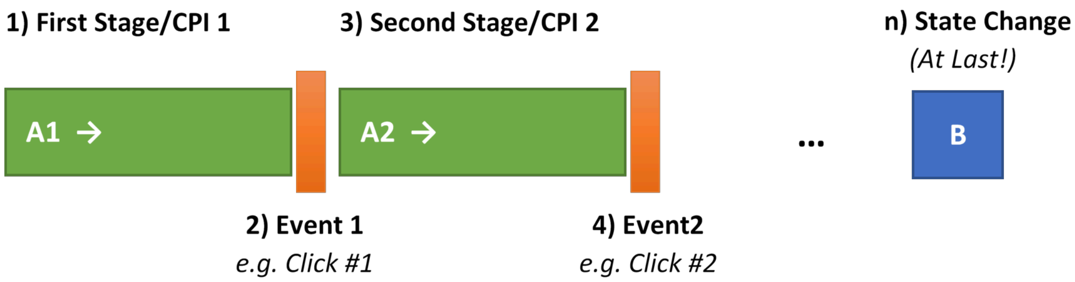


Figure 4.17: *Example of a "Multi-Stage Interactive" SBP. Although only two stages are shown, it is possible to add in more stages in a similar way. Here execution starts by playing the "`A1->`" Cycle Pattern until `Event 1` occurs, at which point the "`A2->`" Cycle Pattern is played until `Event 2` occurs. This process continues on for any successive Cycle Pattern – Event-Interrupt pairs until there are none left. At that point, the State Change (to State `B`) occurs.*

This pattern can be used for several different purposes. One of these is to be included in highlighting techniques used to prevent accidental deployment/invocation of certain commands. For example, this could be used for a highlighting technique for an "interlock" mechanism, where users have to deliberately perform some action multiple times to unlock the mechanism, with the highlight changing colour each time the action is performed (e.g. the highlight has to change from red, to yellow, and then to white for the mechanism to be deployed). Another example could be a highlighting technique used to teach users some new hotkey combination, with each different stage used to indicate that a different part of the combination needed to be pressed.

4.4.3 Cycle Pattern

Cycle Patterns (CP) are the third-tier mechanism for controlling how IHT's behave over time. The main difference between SBP's and Cycle Patterns is that SBP's are more focussed on describing the relationship between user input and state transitions, whereas Cycle Patterns are focussed more on describing how and when Animation Snippets get played.

4.4.3.1 Cycle Pattern Notation and Key Concepts

As with SBP's, Cycle Patterns can have their own graphical notation scheme to make it easier to communicate the construction of different Cycle Patterns. While both SBP's and Cycle Patterns are examples of temporal patterns, a different notation scheme is needed as Cycle Patterns have a different set of concerns.

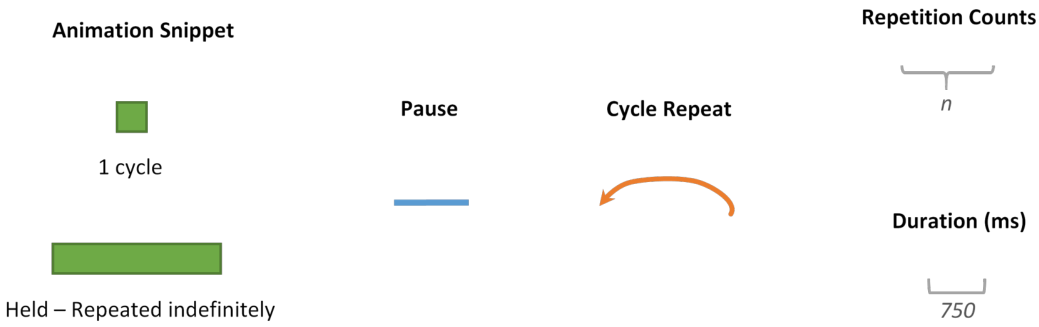


Figure 4.18: *Overview of the element types used for Cycle Patterns.*

Figure 4.18 shows the key notational elements that can be used as part of a Cycle Pattern.

- **Snippets** – These represent one iteration of the animation snippet used. They are shown as square blocks (one for each repeat of the cycle), or just as a long block (if the cycle will be held for an infinitely long time instead).
- **Pauses** – These represent a period of time where no cycle pattern is applied. They are shown as lines linking the blocks together, with the length of the line representing the length of the pause. If no duration indicator is present.
- **Looping Arrows** – These indicate where the Cycle Pattern should repeat, and how many times. If absent from a pattern, this indicates that the pattern should just be played as a one-off animation.
- **Repetition Counts** – These are used when specifying patterns where the snippet needs to be repeated a certain number of times (but more than can be conveniently expressed on a page), or when defining general classes of patterns (e.g. N-Pulses Then Pause)
- **Durations** – These are used for annotating pauses (much like how Repetition Counts are used for sets of snippets) to indicate how long the pause lasts (in milliseconds).

As with State Behaviour Patterns, Cycle Patterns are interpreted from left to right. Each snippet and pause should be executed when encountered. When a curved arrow is encountered (or more specifically, the non-arrow end of such arrows), execution should return to the step that the arrow head points to.

Figure 4.19 shows a second type of diagram ("Lightbulb Plots") that can be used to visualise the temporal effects of Cycle Patterns. It shows how Cycle Patterns would be perceived when used to control the flashing of a lightbulb. Each star glyph corresponds to an animation snippet being played once. Shaded-ellipses shown behind the stars are used to indicate pulses that are sustained instead of being quick flashes; the width of the ellipse represents how long the light stays lit.

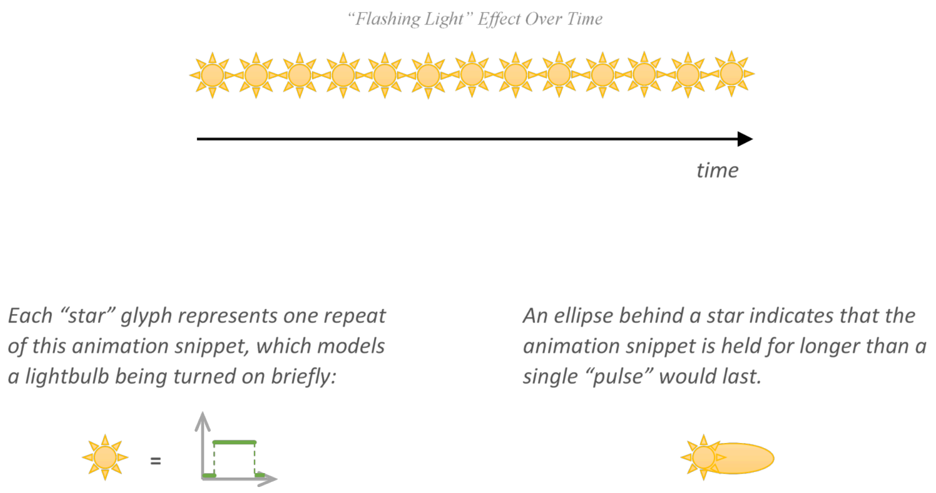


Figure 4.19: *An alternative way of representing the effects of Cycle Patterns. This representation works by showing how the pattern would behave if assigned to a lightbulb, where the intensity of the lightbulb is animated using a short off-on-off animation snippet.*

To visualise the Cycle Pattern depicted, imagine that each star represents a short videoclip showing a lightbulb that is initially off, lights up, and then turns back off again. The longer the region occupied by the lightbulb along the timeline (travelling from left to right along the diagram), the longer the lightbulb stays illuminated. Each of these "lightbulb videoclips" is an Animation Snippet containing an F-Curve with a "Square" shape (Section 4.4.6).

4.4.3.2 Frame Rate, Frequency, Durations, and Number of Repetitions

The interplay between the concepts of "Frame Rate", "Frequency", "Duration of Snippets", and "Number of Repetitions" are important when working with Cycle Patterns, as these concepts concern the speed and length of the Cycle Pattern.

Frequency refers to "potential repeat rate", or the number of times that an animation snippet "could" be played within the space of one second (if it were allowed to be repeated as many times as necessary to fill that time interval). It is also alternatively known as the "speed" of the cycle pattern. This has units of Hertz (Hz) or times per second.

Duration of Snippets (or **Period**) refers to how long an animation snippet runs for when played at the desired speed or frequency. That is, the period of an animation snippet is a measure of time, with units of milliseconds (ms).

To understand how this works, consider the example of an animation snippet which animates the intensity of a lightbulb, going from 'off', to 'on', and then back 'off' again in a sinusoidal way. If we need this snippet to repeat at 1 Hz, the snippet duration is 1000 ms (i.e. 1 Hz = 1000 ms / 1 time). If 2 Hz is required instead, then the snippet duration becomes 500 ms (i.e. 2 Hz = 1000 ms / 2 times). For 3 Hz this is 333 ms (i.e. 3 Hz = 1000 ms / 3 times), and 250 ms for 4 Hz (i.e. 4 Hz = 1000 ms / 4 times). Thus, the period decreases non-linearly as frequency increases.

Number of Repetitions refers to the number of times an animation snippet is actually played as part of the cycle pattern. The important thing to note here is that this is independent of frequency. For instance, it is possible to play a snippet at the frequency/rate of 4 repeats per second, while only actually repeating the snippet twice (i.e. 2 pulses).

Frame Rate refers to the number of "*frames*" shown per second. Based on techniques carried over from traditional film-based systems, computer displays and animation systems work by displaying a stream of still images/snapshots (i.e. "frames") at high speed; each frame is visible for a fraction of a second. Traditional media used the following frame rates (in "frames per second" or "fps"[2]):

- *Film* = 24 fps (or 41 ms per frame)
- *PAL Video* = 25 fps (or 40 ms per frame)
- *NTSC Video* = 30 fps (or 33 ms per frame)

All of these frame rates are above the 15-20 fps threshold needed for viewers to be able to perceive the display as showing smooth/fluid motion [161]. Modern computer displays and most popular graphics engines (notably HTML/Web Browsers, and QtQuick/QML) use frame rates of 60 fps (16 ms per frame) [125, 53]. Thus, Animation Snippets for IHT's are defined to work at 60fps.

4.4.3.3 Examples of Cycle Patterns

In this section, we present examples of the following Cycle Patterns which are frequently used in many IHT's: **Once Only**, **Repeated Infinitely**, **Run Once Then Pause**, and **N-Repeats Then Pause**.

"Once Only" Cycle Pattern

The *Once Only* Cycle Pattern contains only a single element: the snippet itself. It is graphically represented by a single "1 cycle" square, as shown in Figure 4.18. Example uses of this Cycle Pattern include state transition animations and other non-repeating animations.

"Repeat Infinitely" Cycle Pattern

As with *Once Only*, this pattern only contains a single element (i.e. the snippet itself). The only difference between these two is that in this pattern the snippet is just repeated an indefinite number of times. The graphical representation for this is shown in Figure 4.18 as the "Held" rectangle. This particular cycle pattern is therefore best suited for use with the "Hold Until Cancelled" or with the "Multi-Stage" patterns, where the CPI is followed by an event-interrupt block to determine when to stop evaluating.

[2] Frames per Second (fps) is equivalent to Hertz (Hz). For example, 60 fps displays are refreshed at 60 Hz.

"Run Once Then Pause" Cycle Pattern

The "Run Once Then Pause" Cycle Pattern works by alternating between playing the snippet and pausing/not doing anything (see Figure 4.20a).

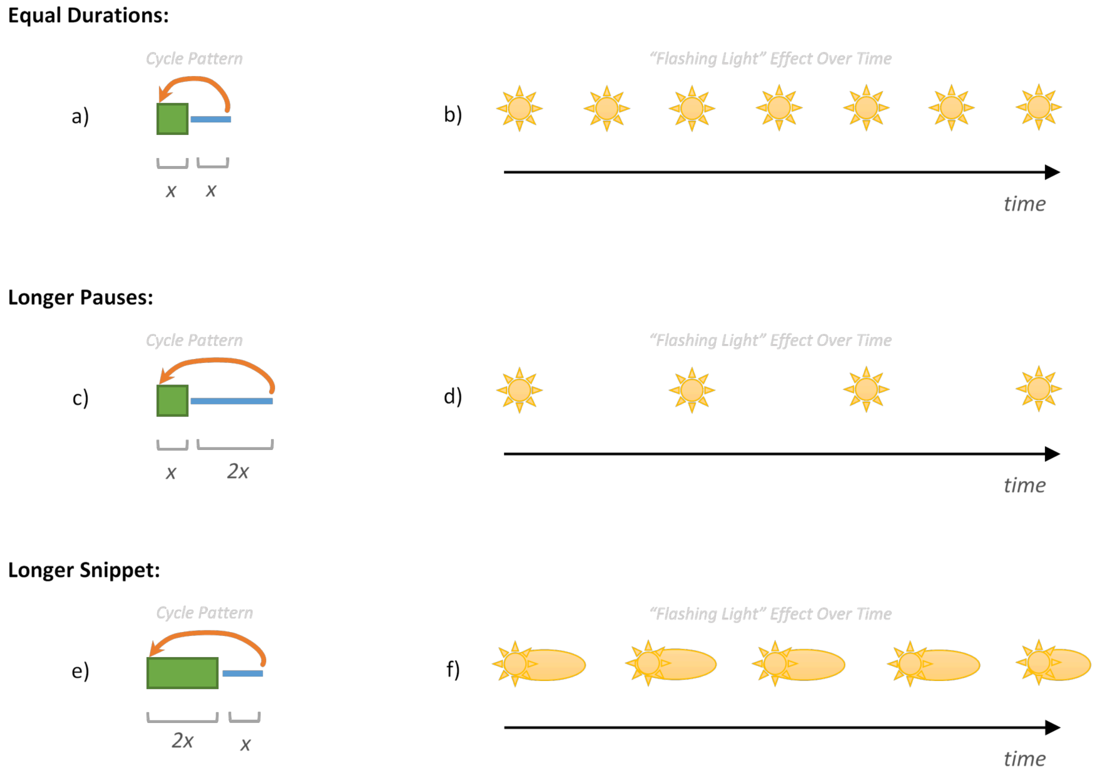


Figure 4.20: *Illustrations of the differences between using different snippet lengths and pause duration ratios for the "Run Once Then Pause" Cycle Pattern.*

The length of the pause can be varied for different effects. For example, if the length of the pause is equal to the length of the snippet (Figure 4.20a), then this creates the effect of a light blinking at a steady rate, especially when applied to a colour or opacity setting (Figure 4.20b). However, if the pauses are longer (Figure 4.20c), the pattern is perceived as being "slower" or less frequent (Figure 4.20d). In contrast, if the pauses are shorter than the snippets (Figure 4.20e), the pattern appears more abrupt/uneven (Figure 4.20f).

"N-Repeats Then Pause" Cycle Pattern

The *N-Repeats Then Pause* Cycle Pattern works by repeating the animation snippet N times (where N is an integer), pausing, and repeating the whole cycle again (Figure 4.21). It is a more general version of the *Run Once Then Pause* pattern (i.e. when $N = 1$, they are equivalent).

Typically, the values of N for this Cycle Pattern are 2, 3, and 4. Examples of these patterns and their effects are shown in Figure 4.21.

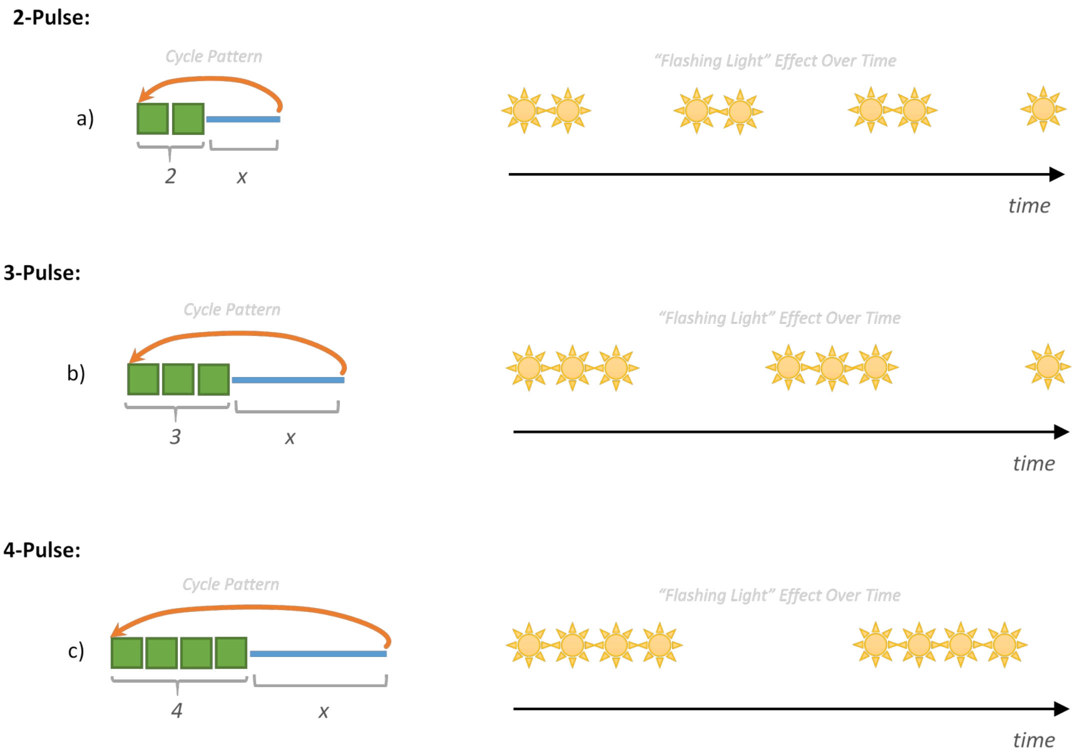


Figure 4.21: *Illustration of the differences between different instantiations of the "N-Repeats Then Pause" Cycle Pattern for different values of N (where N is the number of times the snippet is repeated). Here, the patterns for N = 2, 3, 4 are shown.*

Research Question 3
Is there any reason why "N-Repeats Then Pause" Cycle Patterns are rarely used with more than 4 repeats?

As with the previous pattern, the length-of-pause options discussed also apply here. The main difference between these two is that in order to achieve "equal durations", it is necessary the durations of the pauses must be equal to the time taken to play the snippet N times (as seen in Figure 4.21).

However, not all instances of this pattern use equal durations. Figure 4.22 shows several examples of common N-Repeats Then Pause cycles where shorter pauses are used (a) and one where longer pauses are used instead (c). The version with the shorter pauses is useful for indicating urgency [155], while the version with longer pauses is often used to remind users of something that they may have forgotten about (e.g. it is used in Candy Crush Saga to indicate a suggested move).

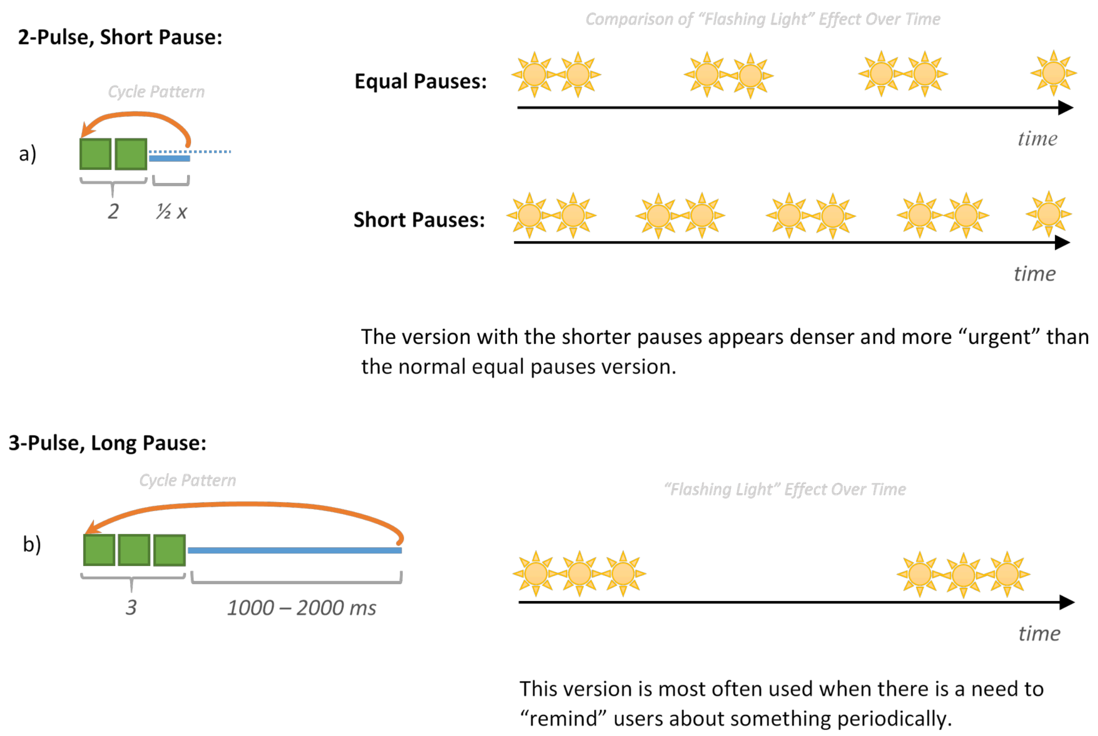


Figure 4.22: *Illustration of two common Cycle Patterns based on the "N-Repeats Then Pause" pattern. These two are notable for having pause durations which are (a) shorter than and (b) longer than the combined duration of the repeats.*

4.4.4 Animation Snippets (F-Curves)

Animation Snippets are a sets F-Curves used to control the behaviour of a highlighting technique's parameters. *F-Curves* (also known as "avars"/"animation variables" [135] or "animation curves") describe how some parameter's value changes over time [105].

Animation Snippets can be characterised along several dimensions: the snippet's place along the "*Static-Dynamic continuum*", and the Symmetry of the F-Curves. Figure 4.23 shows an overview of the design space formed by this pair of dimensions dimensions and how they interact.

4.4.4.1 Static-Dynamic Continuum

The *Static-Dynamic Continuum* concerns how much the property being animated by the F-Curve changes during the snippet. "*Static*" F-Curves are those where the property does not change, while "*Dynamic*" F-Curves are those where the property value changes on most frames. While most F-Curves are quite straightforward to classify on this spectrum (as they fall on either extreme), others are less obvious (e.g. "Step" and "Held" in Figure 4.23).

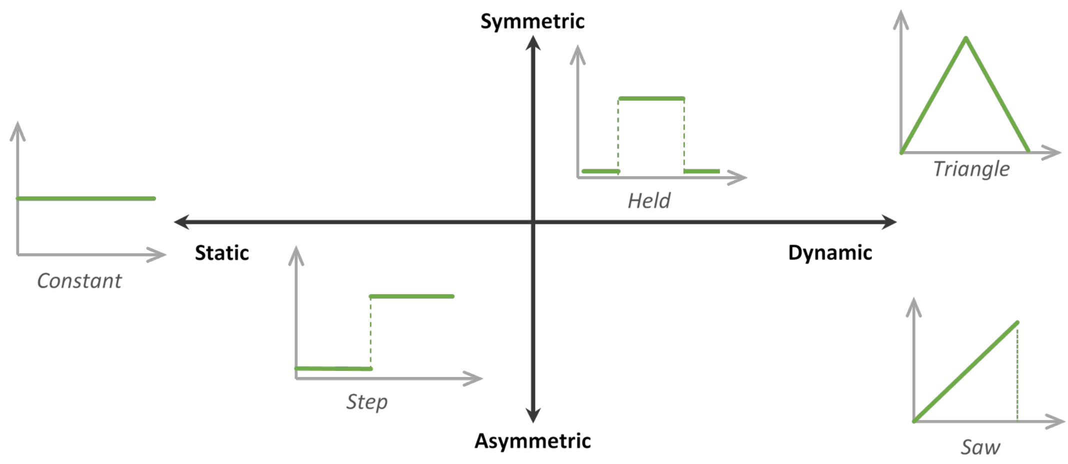


Figure 4.23: Overview of the F-Curve design space, and where common some common techniques fit in relation to the Static-Dynamic axis and the Symmetric-Asymmetric axis.

Although static techniques are more commonly found in user interfaces than dynamic ones, dynamic techniques have become increasingly popular over the past decade [29]. This can be attributed to increases in computing power on consumer-grade devices, and the development of multimedia-friendly platforms such as web browsers, modern operating systems for desktops/smartphones, and UI toolkits supporting animated effects (e.g. QtQuick/QML [53] and Artkit [87]). As a result of these changes, it has become easier to develop and deploy dynamic techniques, whereas technical limitations previously meant that designers were limited to using static techniques in most cases.

4.4.4.2 Symmetry in Animation Curves

The symmetry of F-Curves can be considered in several ways:

1. **Endpoint Symmetry** – *Symmetrical* F-Curves start and finish on the same value, while *Asymmetrical* ones do not.
2. **Spatial Symmetry** – F-Curves oscillating around a baseline/home value (e.g. zero) are "Spatially Symmetrical" if the amplitudes of the curve above and below the baseline value are the same. However, if the amplitudes are different (e.g. the value rises higher above the baseline than it descends below it), the curve is "Spatially Asymmetrical".
3. **Temporal Symmetry** – F-Curves are "Temporally Symmetrical" if the time covered by the first $n/2$ keyframes is the same as that covered by the last $n/2$ keyframes. For instance, an F-Curve with 5 keyframes at times $t1, t2, ..., t5$, would be Temporally Symmetrical if the time difference between the first half (i.e. $t3 - t1$) was equal to that of the second half (i.e. $t5 - t3$).

Symmetrical F-Curves are most useful for **N-Repeats Then Pause** Cycle Patterns (where the F-Curve is repeated multiple times), as they restore the animated property back to its original value. In contrast, Asymmetrical F-Curves are more commonly used for transitioning between states. For example, Asymmetrical F-Curves can be used in the *Initial Onset* or *Dismissal* states to create a gradual transition between the highlighted and unhighlighted states (e.g. fading in/out the opacity of an item instead of abruptly toggling its visibility).

4.4.4.3 Keyframe Animation and Parametric Curves

Most animation curves are defined using keyframe animation. Keyframes are time-value pairs defining the value that a property should have on a particular frame. They are not typically defined on every frame (except where fine control is needed), with frames "in-between" each pair of keyframes computed by *interpolating* between them or by using *easing* equations (see Section 4.4.5).

Curves not defined using keyframe animation are instead generated parametrically. That is, a mathematical function (e.g. $f(T) = \alpha T + \beta$, where $x = T$, $y = f(T)$, and α and β are parameters used to control the shape of the generated curve) is used to determine the value of the curve at each frame. Examples of functions often used for this purpose include trigonometric functions (e.g. $sin(x)$, $cos(x)$, $sin(x)/x$) and polynomials (e.g. $Ax^2 + Bx + C$).

The following two sections (Section 4.4.5 and 4.4.6) discuss in more detail how F-Curves can be constructed.

4.4.5 Keyframe Interpolation for F-Curves

Interpolation and Easing Equations are used to define how a parameter's value changes over time between between two keyframes (k_a and k_b). In this section, we will refer to a the value on the first keyframe (k_a), and b as the value of the second keyframe (k_b). It should be noted that a is not always less than b. For example, consider an F-Curve with a `low-high-low` shape:

- If k_a is the *first* keyframe (low) of this F-Curve, and k_b is the middle keyframe (high), then $a < b$.
- However, if k_a is the *second* keyframe (high) of this F-Curve, and k_b is the last keyframe (low) then $a > b$.

4.4.5.1 Interpolation Techniques

Three interpolation methods are commonly available in most animation software:

- **Constant Interpolation** – The value of this first keyframe is held until the second keyframe is encountered.
- **Linear Interpolation** (lerp) – Linear Interpolation blends between the two keyframes. The relation contribution of each keyframe is determined by using the proportion (t) of the time travelled between the keyframes by the current frame. That is,

$$Lerp(t) = (1 - t) \times k1_{value} + (t) \times k2_{value}$$

- **Bezier Spline Interpolation** – Instead of blending between keyframes, t is used to look-up a point on a cubic Bezier spline instead. For the highest levels of user control, this technique adds two extra vertices – one after the first keyframe, and one before the second keyframe – to define two tangents to have some control over the shape of the curve as it leaves the first keyframe and arrives at the second.

Interpolation/Easing types can be set differently for each curve segment (i.e. between each pair of keyframes). So, it is possible to use Constant Interpolation for sections where there are no changes, followed by Bezier interpolation for a more complex segment.

4.4.5.2 Robert Penner's Easing Equations

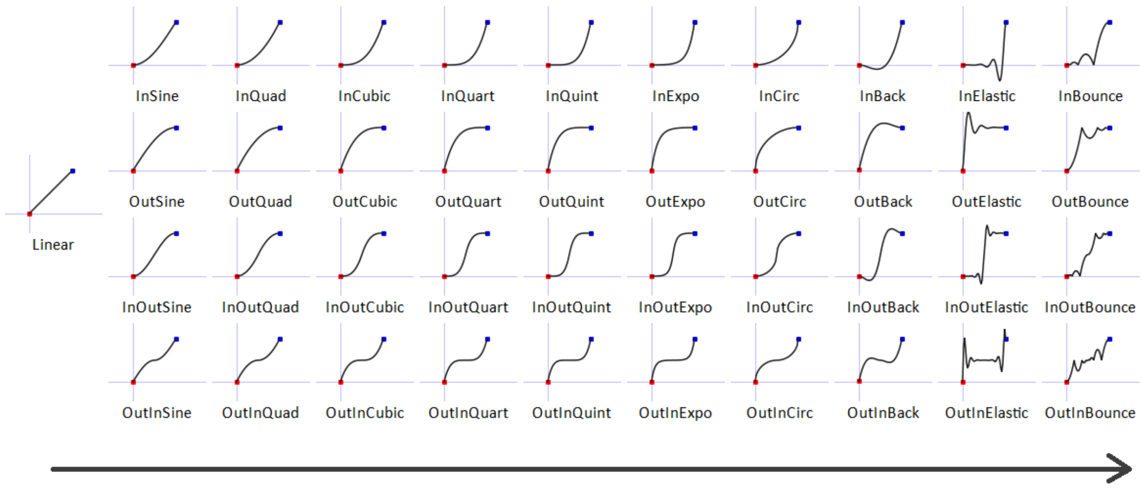


Figure 4.24: *Example graphs of easing equations. These show the change in the value of a parameter (vertical-axes) over time (horizontal-axes). Red dots indicate the first keyframe, and blue dots indicate the second keyframe. Techniques are ordered by strength of effect, from weakest (left) to strongest (right), as described by Penner [129].*

Robert Penner's "Easing Equations" [129] are a well-known set of equations for computing F-Curve values between keyframes (see Figure 4.24). They were first introduced in Chapter 7 of Penner's book on ActionScript programming [129], and have since been incorporated into many basic user interface and graphics frameworks in widespread use, including Qt [52] and CSS3 [146]. As seen in Figure 4.24, effects can be selectively applied to one end of the transition (e.g. "in" or "out"), or to both ends (e.g. "in-and-out", or "out-and-in").

There are two types of easing equations:

- **Transitions** – These describe how the value of a parameter changes from value a to value b. Not all of the easing transitions strictly interpolate between the values of a and b. Some transitions such as the "Back" effect extend beyond the bounds imposed by the keyframes, creating over/undershoot effects.

- **Physics-Based Special Effects** (e.g. "Bounce" and "Elastic") – Some easing equations produce "special effects" or stylised approximations of the way that physical objects behave when manipulated. For example, the "*Bounce*" effect simulates a rigid object bouncing on a hard surface (e.g. a "bouncing ball"), and is most suited to situations where an element (e.g. a sliding pane) needs to behave like it is colliding with another element (e.g. the screen edge) in a physically-plausible way. These, physics-based effects are often used for animating transformations (like translation and rotation for *Bounce*, and rotation and scaling for *Elastic*).

4.4.5.3 Spacing and Rates of Change

The slope (or rate of change) along an F-Curve can affect how users perceive a transition [128]. For example, consider an icon travelling across the screen over several seconds. On each frame, the icon appears at a different position on screen. When the icon moves quickly, the positions of the icon on adjacent frames are more sparse, while when the icon moves slowly, the positions are closer together. Animators refer to this concept as "spacing" [104].

Dragicevic et al. [59] conducted an empirical study examining how object tracking performance varied when different transition types were used, in order to determine whether *Slow-In/Slow-Out* (i.e. endpoints are emphasized) were as effective as commonly assumed. They compared this to *Constant Interpolation* (i.e. frames are equally spaced, so nothing is emphasized), *Fast-In/Fast-Out* (i.e. the middle is emphasized), and *Adaptive* (i.e. the frames with the highest visual complexity are selected to be slowed down) [59].

Slow-In/Slow-Out was found to outperform all other techniques [59], with a "significantly higher" tracking accuracy [59]. Constant speed was the second best technique overall, with Adaptive speed only effective when the visual complexity occurs towards the start/end of the motion (i.e. when Slow-In/Slow-Out would work equally well) [59]. From these findings, Dragicevic et al. concluded that it is more important to allocate more of the frame budget to the endpoints so that users could anticipate the motions than it is to slow down complex segments so that they could be seen more clearly.

> **Design Insight 30**
> *Users find it easier to track moving objects when they have more predictable motions (where more frames are dedicated to the start/end of a motion) [59].*

4.4.6 Parametric F-Curves: Examples of Commonly Used Animation Snippets

Figure 4.25 shows 6 examples of animation snippets commonly used as part of Cycle Patterns. The examples presented here have been chosen as they are often used in highlighting techniques. We refer to these as being "Parametric F-Curves", as the shape of these can be modified by changing the relative timing of the keyframes (i.e. $k1$, $k2$, …, kn), and the minimum/maximum values of the keyframes (denoted as α and β).

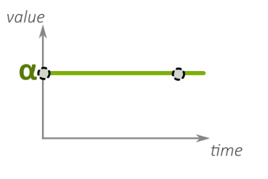
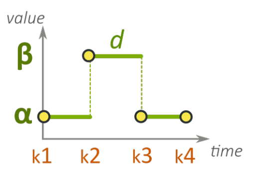
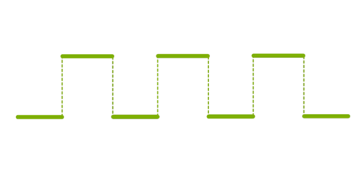
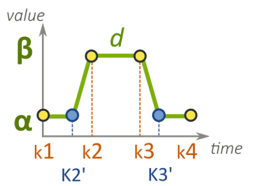
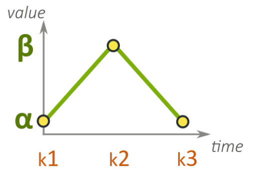
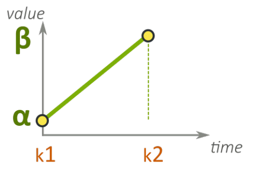
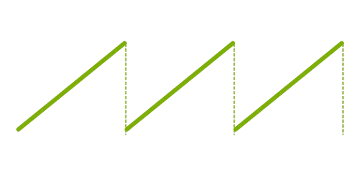
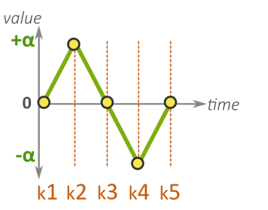
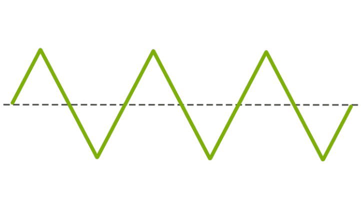
	Name	Graph	Keyframes	Values	Graph (x3)
A	Constant		N/A	α	
A-B-A (Pulse/Flash)	Square		k1, k2, k3, k4	α, β	
A-B-A (Pulse/Flash)	Gradual On-Off		k1, k2, K2', k3, K3', k4	α, β	
A-B-A (Pulse/Flash)	Triangle		k1, k2, k3	α, β	
A-B	Saw/Linear		k1, k2	α, β	
Oscillate	Sine		k1, k2, k3, k4, k5	+/- α	

Figure 4.25: Overview of some common animation snippet shapes. The horizontal axis on each graph represents time, and the vertical axis represents the parameter values. Coloured dots on each curve-line indicate keyframes. The coloured bars on the left-hand side of the table indicate which techniques can be used for similar purposes.

The usage of each snippet type shown in Figure 4.25 are as follows:

- **Constant** F-Curves are implicitly used for all "static" parameters (i.e. parameters whose value does not change over time).
- **Square** F-Curves are used to animate pulsing/flashing effects when the parameter can only have binary values (i.e. On/Off). The name comes from its resemblance to the "Square" waveform shape [171] from audio/signal processing.
- **Gradual On/Off** F-Curves are a variation of *Square* curves. However, instead of abruptly transitioning between the minimum and maximum values (i.e. α and β), this variant features gradual transitions. This makes it similar to incandescent light bulbs, where the light output increases gradually as the filament heats up. They are again best suited to flashing effects, where the maximum value is held for a period of d seconds.
- **Triangle** F-Curves are used to animate pulsing effects (e.g. a light bulb steadily turning on and off, or a button expanding and returning back to its original state as if it were breathing). It can be thought of as a special case of Gradual On/Off, where d is zero (i.e. the minimum/maximum values are not held). The name for this pattern comes from its resemblance to the "Triangle" waveform shape [172] from audio/signal processing.
- **Linear** F-Curves describe a transition between two values (i.e. between α and β); this is most useful for animating state transitions (e.g. it can be used in the *Initial Onset* and *Dismissal* states to transition between the *Normal* and *Highlighted* states).
- **Saw** F-Curves are formed by repeating a *Linear* F-Curve snippet multiple times. The resulting curve looks like the jagged teeth of a saw, resulting in this pattern being known as the "Sawtooth" waveform [170]. It can be used for effects where the animated item should "jump back" to the starting point every time the animation repeats, or it can be used for cyclic-offset animations (e.g. animating a texture travelling around a window border by offsetting the texture coordinates).
- **Sine** F-Curves are used to animate a parameter oscillating around a value. For example, a shaking movement (i.e. the highlighted item moves side to side around its starting point) can be created by using a F-Curve of this type.

4.5 Reevaluating the Highlighting Techniques Landscape

Highlighting techniques described in the prior HCI literature can be re-framed in terms of our PCCH design framework. Some examples follow.

Harrison et al.'s work on *Kineticons* [81] showed examples of transformation and deformation effects applied to the *Base* layer of widgets. The parameters of these techniques were animated to create the effect for each kineticon. For example, "Whole Icon Wave" (in which the icon rolled side to side around its midpoint) was created by using a sine-wave F-Curve to animate the rotation angle of the *Base* layer (and its children). The "Bounce" kineticon (which was similar to the bouncing effect applied to icons in the Mac OSX dock) was also created by using a sine-wave F-Curve, but applied instead to the `Y-Offset` (i.e. vertical translation) parameter of the *Base* layer. Some effects were also more complicated, such as how the positions of the corner vertices of the deforming free-form deformer were keyframe animated used to create the “X-Cross” kineticon (where the icon was distorted to from an ‘X’ shape) or the “Running” kineticon (where the corners of the icon are animated as if they were the arms and legs of a character).

Harrison et al.’s work on *Flashing Patterns for Point Lights* [80] examined different ways of controlling the brightness/intensity of LED’s (such as the indicator light on a smartphone) to communicate information such as low battery state, and other simple state information. The techniques presented were examples of different Animation Snippet and Cycle Patterns that could be used to animate a parameter. In their paper, the parameter in question was the “energy” or “brightness” control on an LED. But, these patterns could equally apply to any other parameters (e.g. position or colour of a widget).

The value of our design framework extends beyond animated effects. For example, Fitchett et al.’s *Finder Highlights* [67] used highlights to help users quickly locate files and folders they are likely to want to access, as predicted from past revisitation and access patterns. Soft-edged yellow circles were drawn behind the file/folder icons being highlighted. These yellow highlights were examples of transient entities that may be drawn in the *Underlay* layer; another example is the blue-selection rectangle that is drawn around/behind selected items in standard file browsers [63, 75, 58, 97].

Agapie et al.’s experimented with changing the appearance of the borders of a text field [10] as a hint to the user that they should enter a longer and more detailed search query. When the query string was too short, the borders were red. If the query string was of a sufficient length, the borders would change to a blue colour instead. This was an example of a highlighting technique with state-dependent behaviour: if we assume that the “not long enough” state corresponded to the “Highlighted” state (as we are trying to draw the user’s attention to the fact that their input is invalid), then the “is long enough” state would correspond to the “Normal” state. Each of these states has its own distinctive appearance and behaviours, described using different State/Cycle/Animation patterns as appropriate.

5 Summary of Foundations

Highlighting techniques are visual communication techniques used to draw user attention to salient items. A number of different techniques have been developed, and are used for a diverse range of applications. The effectiveness of a highlighting technique can be represented in terms of how noticeable and distracting they are. Other metrics of highlighting effectiveness which may be useful in some situations include the accuracy of detection (True Positive Rate (TPR) and False Positive Rate (FPR)), as well as subjective experience measures. Using these metrics of highlighting effectiveness, it becomes possible to make meaningful empirically-driven comparisons and decisions about whether one highlighting technique is better than another for a given scenario.

5.1 Insights from Underlying Human Factors

Various human factors affect the effectiveness of highlighting techniques. Change Blindness (CB) and Inattention Blindness (IB) are two of the biggest problems that effective highlighting techniques have to overcome: *Change Blindness* occurs when a physical distraction/disruption making it harder for the observer to notice changes, while *Inattention Blindness* occurs when the observer was too absorbed in other tasks (e.g. perhaps because the current task had become overly routinised).

CB and IB occur due to the limitations of the human visual system and attention processing pipeline. The structure of the eye means that colour and detail can only be detected in a very small region in the center of the visual field. Thus, visual acuity is lower away from the fovea (i.e. in peripheral vision). Our attention is also a very limited/scarce resource, which can only be directed towards a single target at any point in time. Therefore, the job of highlighting techniques is to redirect the user's attention towards information that they should be made aware of. However, highlighting technique should provide a good amount of utility to the user in exchange for the effort that they need to expend to attend to the highlight, (i.e. highlighting strength should be proportional to the importance/utility of the information being highlighted).

5.2 Summary of Our PCCH Framework based on the Construction and Control of Highlights

Our PCCH design framework provides a structured system for consistently and precisely describing the construction and behaviour of highlighting techniques. Compared with prior frameworks, PCCH has a broader scope, is more precise, and serves as a form of design guidance. It also has a more practical and pragmatic focus – PCCH adopts and adapts many industry-standard concepts from computer graphics engines and UI toolkits, and combines these with a set of novel highlighting-specific constructs to provide the HCI community with a powerful new framework for describing highlighting techniques in terms of how they are constructed and controlled.

Using our PCCH framework, Interactive Highlighting Techniques (IHT) can be described in terms of the layers of Visual Elements used to construct them, the Pixel-Level and Object-Level manipulations applied to those elements, and how the parameters exposed by those elements and manipulation effects behave over time. The behaviour of an IHT over time can be described in terms of four levels of control structures: the State Transition Model, State Behaviour Patterns, Cycle Patterns, and Animation Snippets/F-Curves. These control structures work by describing how key parameters exposed by the visual elements and their manipulations change/behave. Using this parameterisation, we can not only construct and describe highlighting techniques in a more systematic way, but it also becomes possible to identify novel ways of combining visual effects to create a new highlighting technique.

5.3 Review Summary for Prior Studies of Highlighting in HCI

5.3.1 Prior Frameworks and Applications of Highlighting

There have been several prior attempts at developing design frameworks for highlighting techniques [37, 106, 119]. All of these are based on a partitioning of the design space popularised by the Gestalt Principles [102]. However, frameworks based on the Gestalt Principles are less suitable for describing highlighting techniques in an unambiguous way that designers or their support tools can easily process, analyse, and use when predicting performance (in the form of Noticeability and Distraction) with an unknown technique.

Highlighting techniques are used for a wide variety of purposes including helping the user to: 1) navigate the interface (e.g. Scented Widgets [173], or with Navigation Hints [67]), 2) manage information overload (e.g. in menus [64]), or 3) ease the novice-to-expert transition (e.g. Ambient Help [116], and Heatmaps of Useful Features [117, 143]).

5.3.2 Noticeability and Distraction

The tradeoffs between the noticeability and distraction characteristics of different highlighting techniques are currently not well understood. While many studies have focussed on measuring noticeability, relatively few have explored the issue of distraction and its effects and relationship to noticeability.

Noticeability has been well studied in the psychology and HCI literature (particularly in the former). Compared to distraction, it is arguably easier to define and measure – as a result, many well established experimental procedures exist for measuring it under controlled laboratory settings [28, 57, 86, 99]. Noticeability is also arguably more important: a highlighting technique that is unnoticeable is unlikely to be of much use, whereas a highly distracting technique may still be used if that technique is the only technique that can capture the user's attention as required.

Distraction has been studied less frequently than noticeability. However, recent work by Gallivan and Chapman [72] and Moher et al. [121] have identified promising methods for measuring the distraction effects of different visual effects based on analysing the three-dimensional paths traced by participants acquiring targets on a gridded pattern on a screen.

5.4 Emergent Guidelines for Effective Highlights

We propose the following set of design guidelines for creating effective (i.e. easily noticeable) highlights based on insights from the prior literature on human factors and noticeability.

1. **Consider the importance of interrupting the user when choosing highlighting techniques, and choose an appropriate time (e.g. between tasks) for interrupting**. Different pieces of information have different levels of importance and urgency. The "strength" (i.e. intensity of parameter values) of the highlight chosen should correspond to these priorities:

   - ***Urgent and Important* information should use the *highest* strength techniques.** Examples of "high strength" techniques include those using high repeat frequencies (e.g. 3-8 Hz), large motion amplitudes (e.g. 15-30 px movements), and/or high colour contrasts (e.g. bright red on black, or light yellow on dark blue).
   - ***Urgent but Not/Less Important* information should use *noticeable but less distracting* techniques.** Examples of these include techniques using more "moderate" values for each parameter (e.g. 2-4 Hz frequencies, 5-10 px amplitudes, or `Orange` on `SkyBlue` [3]).
   - ***Non-Urgent but Important* information should use *low* strength techniques which are minimally distracting.** Examples of "low strength" techniques include those using low repeat frequencies (e.g. 0.1-2 Hz), small motion amplitudes (e.g. 1-5 px movements), or low colour contrasts (e.g. `#DDDDDD` vs `#BBBBBB`).

2. **Temporal Effects should be used if it is necessary for the user to notice the highlight in a timely manner**. This is because temporal effects (e.g. moving, shaking, wobbling movements or flashing) are more readily detected (i.e. the user reacts quicker) than static graphics. Therefore, temporal effects should be considered when it is important that the user notice an alert, or when indicating causal relationships (i.e. to reinforce the idea that highlight A appeared at point B because the user performed action C). Examples of situations where temporal effects could be useful include using highlights to teach the user a shortcut for a commonly used tool, or to remind them to pay attention to a critical indicator needed ensure safe completion of the action.

3. **Initial events occurring nearer to the user's focal point are more likely to be noticed**. When it is important that the user's attention is immediately drawn towards some distant point the interface (e.g. on the opposite side of the screen, or near the edges on a toolbar/status bar), the effect should initially appear near the user's current focal point (e.g. typically near the cursor, or if eye tracking is possible, the current fixation). This is because vision decays in peripheral vision, and it becomes harder for the user

to detect changes further out (especially if items are closely spaced, as predicted by Bouma's Law).

4. **Use large visual elements for more noticeable effects**. Larger items (up to a certain point) are easier for the user to notice, especially in peripheral vision [99]. For example, consider the difference between the blinking caret/cursor (i.e. 1×20 px line/box) and a flashing banner ad (e.g. a 320×240 px or larger box). Therefore, visual elements which appear on the *Underlay/Overlay* layers or are manipulated to create a highlighting effect should be large to be noticed more easily.

5. **Provide more than one opportunity to notice the highlight**. Change blindness may mean that if a highlight only appears once, the user may have missed the initial appearance of the highlight, and is thus unaware that it exists. Therefore, the "*Reminder*" state should be used to provide periodic reminders that there is still information to attend to.

5.5 Avoiding Negative Impacts of Highlighting

The negative impacts of highlighting (e.g. such as distraction) may occur when using highlighting techniques, or when design decisions inadvertently cause certain elements to act as highlights.

5.5.1 Avoiding Negative Impacts When Deploying Highlighting Techniques

Highlighting effects can have negative impacts on the user experience when deployed due to problems such as a mismatch between user needs and what the designers are trying to achieve, as well as the use of ineffective or overly strong/forceful highlighting techniques.

A mismatch between the user's needs and what the designers are trying to achieve results in the user being bombarded with information that they do not care about or want to know about (e.g. banner advertisements). A symptom of these mismatches is when the sampling payoff (i.e. the opposite of the Expected Cost, as the Expected Cost represents the cost of *missing* the highlight) is low. For example, a highlighting technique which provides useful/important information but which is very difficult to detect offers a low sampling payoff. This is because most of the time, the user is unable to detect the highlight (and has trouble detecting it even if it is there), implying that the user must expend quite a lot of effort. Since the highlight is very important when noticed, the user needs to be constantly be on the lookout for the highlight as they need to be very careful to be able to detect the highlight.

Ineffective highlighting techniques like this cause users to adopt "paranoid" monitoring strategies. A paranoid monitoring strategy means that the user often ends up "oversampling" the various channels/status indicators in the hope that there may be a change to notice. However, such behaviours also correspond with a higher number of false positives (i.e. false alarms such as the "phantom buzzing" phenomenon) and elevated stress levels (as the user needs to be alert and vigilant to avoid missing the highlight, and suffering the consequences of that).

Overly forceful highlighting techniques have the opposite problem in that they prevent the user from concentrating on their primary task, which results in annoyance from the constant interruptions.

5.5.2 Avoiding Accidental Highlights

A second type of negative impact of highlighting is when designers accidentally or inadvertently creating highlighting techniques on items that were not meant to be highlighted. An example of an inadvertent (and distracting) highlight is applying a bold yellow colour or texture to a large navigation sidebar positioned close to (within 1 cm of) the text of an article.

To avoid making highlighting techniques too distracting, a useful starting point is to consider the design guidelines for "effective" techniques from the reverse perspective. For example:

- **Animated graphics are more easily noticed than static graphics** – This can be re-interpreted to mean that animation on less-important items *should be avoided* to lessen the chance of accidentally attracting the user's attention.
- **Items located closer to the focal point are noticed more readily** – This can be re-interpreted to mean that less-essential elements should be placed further away from the primary task area, even if this may lead to a less "balanced" visual design. Important items should still however be moved closer to where the user is likely to be concentrating.
- **Large items are noticed more readily in peripheral vision** – This can be re-interpreted to mean that less important items should be smaller, to avoid drawing attention to themselves. For example, sidebars could be made narrower (and/or should be kept further away from the contents of a document/article).
- **Higher contrast items are noticed with greater ease** – This can be re-interpreted to mean that less important items should have less colour contrast, so that they are not that likely to attract the user's attention. Caution is needed as "less important" is context dependent: an individual menu item may be unimportant when the user is concentrated on the content in the middle of the screen instead, but when the user is trying to locate commands (e.g. the mouse is over or near that region), it is important for the user to be able to quickly and easily distinguish between different icons. In this regard, aspects of modern "flat UI" design trends such as "monochrome icons" are problematic.
- **Unique features create pop-out effects** – This can be re-interpreted to mean that any "unique" visual features (such as a salient colour not used elsewhere, a pattern, or some other form of visually salient features) should be avoided if an element should not draw attention to itself.

5.6 Research Opportunities

5.6.1 Characterising Highlighting Technique Effectiveness

There is a need for a systematic set of empirical studies on characterising how effectiveness metrics (such as noticeability and distraction) vary for different highlighting techniques as their parameters are manipulated. Our parametrisation of the highlighting technique design space provides a structured way of approaching this task. It also provides a vocabulary for documenting these findings in a way that can help the HCI community as a whole to build up, share, and reuse our knowledge of the design space more effectively than has is currently possible.

In particular, the prior research hints at several potential interactions which need to be investigated in more detail, such as:

- Which classes of technique are more effective than others? Which are the least effective?
- Is there any interaction between frequency and amplitude of movement, if so, how these interact for commonly used frequencies and amplitudes?
- Is there any interaction between amplitude and size of items? What about frequency and size?
- Do effects repeating at high temporal frequencies (4-8 Hz) provide any significant benefit over lower frequencies (1-3 Hz)?
- Is there any correlation between frequency and perceptions of urgency or other measures?
- How do these techniques perform in aggregate? That is, when the number of highlighted items $n = 2, 3, 4, \ldots$ increases, what effect does this have on the noticeability and distraction of the highlighting technique? (i.e. Does the technique interfere with itself? How does it interact with other techniques that may be present?)
- Does colour and/or background contrast affect highlighting effectiveness, and how (especially in relation to the other parameters)?
- What interactions exist between eccentricity and all the other parameters (frequency, amplitude, size, contrast)?
- Why are animations usually only repeated 2, 3, or 4 times only, and not 5 or more?
- How do these techniques perform after repeat exposure? Are there learning effects, and how severe are they?

5.6.2 Developing New Techniques and Tools

As part of our discussion of how highlighting techniques can be constructed, we identified several promising directions for further investigation such as ways of introducing random behaviour into highlighting techniques (so that the appearance of a highlighting technique is less predictable, and hence, less likely to get ignored by users over time), as well as a multi-stage "interlocks" mechanism which may be useful in some cases.

Another contribution of our control and construction framework was to provide a broad roadmap of ways in which different effects could be developed and combined together to achieve novel techniques. There are many possibilities here that have yet to be explored by the HCI community which this parametrisation opens up.

Other sources of design inspiration are the studies of underlying human factors. In particular, there is a need for techniques which combat Change and Inattention Blindness. For example, possible directions for future research includes attempts to overcome normalisation/adaptation to repetition, techniques which position highlights in places where they are more likely to be detected by the user (e.g. having effects that start closer to the user's focal area instead of appearing in peripheral vision [91]).

There are also opportunities for the development of computer-aided design tools to provide objective guidance for highlighting design. The role of such tools is *not* to replace skilled designers, but rather to assist them. Rosenholtz et al. [140] argue that simply creating sets of design guidelines for designers to follow is insufficient for helping designers make the most of research findings. Instead, computer-aided tools may be better suited to handle all the subtle interactions between different effects [140].

Part II

Methods, Analyses, and Studies

6 A Method for Measuring Noticeability and Distraction

6.1 Introduction

The previous chapters covered the foundational knowledge necessary for understanding the design space for interactive highlighting techniques, and reviewed the prior literature on methodologies for measuring and comparing their effects. As discussed in Chapter 3, there are many opportunities for developing improved methods for analysing and comparing interactive highlighting techniques. This chapter presents the high-level concepts behind our methodology for experimentally analysing and comparing the noticeability and distraction of highlighting techniques. The following chapters (Chapter 7 and 8) cover in more detail the concrete instantiations of these principles.

6.2 Objectives

The key objectives of our experiment methodology were:

1. **Noticeability and Distraction** – To empirically measure both the noticeability and distraction characteristics of different highlighting techniques, within the same experiment procedure, without requiring separate passes or radically different procedures to do so;
2. **Measurable, Reliable, and Sensitive** – To obtain objective measures of noticeability and distraction that are sufficiently sensitive to detect reliable differences between these characteristics when the intensity of highlighting techniques are manipulated;
3. **Pragmatic** – A method that is easy for the HCI community to understand and practically deploy to quantify the characteristics of a pair (or set) of highlighting techniques;
4. **Externally Valid** – A method that has some relevance to HCI applications of highlighting techniques.

The first objective concerns the structure of studies using this method, and the conditions that need to be included in those studies; it differentiates our method from prior HCI studies, where distraction was evaluated separately as post-hoc subjective experience measures. The second objective concerns the quality of performance metrics used to quantify noticeability and distraction, ensuring that they are actually effective at capturing nuances of the effects they target. The third and fourth objectives are aimed at ensuring that our method will be of use to the HCI community (and in particular, for designers seeking a way to evaluate the suitability of their design ideas [140]).

6.3 Key Principles

This section describes the key principles underlying our experiment methodology. To serve the objectives identified, the method needed to be able to objectively measure both noticeability and distraction, while being practical to implement.

6.3.1 Measuring Noticeability

Noticeability is relatively simple to measure, with many obvious solutions available. Measures of noticeability are based on the following key principle:

> **Key Principle 1**
> *Noticeability is a measure of* ***how long after the appearance of a highlight*** *it takes participants to complete some action/task signalling that they detect it.*

This leads to several options. In Studies 1 and 2, we chose the following methods for measuring noticeability:

1. **Click on the Highlighted Item** (Study 1) – This method is best suited to cases where there are a large number of candidate items. By forcing participants to click on the highlighted item, we can determine two things: (a) how well they were able to identify the highlighted item, (b) confirm that they had actually responded to the intended item.
2. **Press a Button** (Study 2) – This method is more suitable when there is only a small number of candidate items, and we are more interested in analysing when participants first become aware of a highlight. It assumes that participants will always correctly identify the intended item. However, with sufficiently strong techniques, this concern may not be an issue.

6.3.2 Measuring Distraction

In this thesis, we define "*Distraction*" to be a measure of the "*amount of performance degradation*" experienced by participants when performing their task(s). That is, as distraction increases, task performance decreases (e.g. it takes longer to complete a task, or there is a higher level of "error" in their responses). This leads to a simple insight about how distraction can be measured:

> **Key Principle 2**
> *Distraction can be measured by* ***comparing the difference*** *between task performance when* ***highlights are present*** *versus when* ***highlights are absent****.*

It is possible to use techniques inspired by both the "Path Deviation" (Section 3.2.2) and "Dual Task" (Section 3.2.1) paradigms to do so, as we demonstrate in Chapters 7 and 8.

6.3.3 Nature of Tasks

To satisfy the key objectives of our methodology, participants needed to perform a large number of trials in order to cover all the required conditions (i.e. at different levels of intensity, with more than a single sample/repetition of each condition, and for both noticeability and distraction) for each of the highlighting techniques. However, there are limits to the number of trials that participants can be subjected to before they become fatigued (i.e. around 150, as found during early stage pilot testing – see Appendix A), and the amount of time available to complete the experiment (i.e. 30 minutes or less) to keep the time commitment reasonable.

To ensure that the experiment is "practical" for the HCI community to use and "externally valid" (i.e. objectives 3 and 4), we also enforce the additional constraint that tasks should be performable using "commonly available" computing equipment. By "commonly available", we are referring to easily obtained (i.e. "off-the-shelf") equipment that users conduct most of their daily computing tasks on (e.g. keyboards, mice, touchpads/touchscreens, and stylus-based input devices). Non-compliant equipment includes custom-built input devices (e.g. single button keyboards), motion-tracking markers [72, 121], and physiological sensors (e.g. Galvanic Skin Response [168] and Electroencephalography [164]). Note however that the use of an eyetracker in Studies 1 and 2 does not violate this constraint, as it is possible to conduct the experiments without the use of an eyetracker (i.e. the eyetracking data was only a secondary metric used to supplement the primary performance metrics).

These requirements lead to the following principle governing the nature of tasks in these studies:

> **Key Principle 3**
> *Tasks need to be short and simple to perform, and should ideally be able to be performed using commonly available computing equipment.*

To ensure that tasks were "short and simple to perform", they needed satisfy two conflicting criteria:

1. **Sufficiently easy to quickly perform** – Tasks should take ***less than 1 minute*** to perform. Longer tasks would result in excessively long experiments (resulting in increased fatigue and greater difficulties recruiting participants), and/or the number of conditions would have to be reduced (i.e. the method is less viable for analysing and comparing multiple techniques). Also, note that "sufficiently easy" means that most participants should still be able to attain reasonable levels of performance in the task.

2. **Sufficiently difficult that participants remain engaged, and/or need to pay attention to the highlights** – We discovered during pilot testing (Appendix A) of the methods presented that if the task was "too easy", the experiment would not be sensitive enough to measure any differences between highlighting techniques, as participants could ignore the highlights completely. Thus, to increase task engagement, the difficulty of the tasks had to be made harder (e.g. the target cueing mechanism needed to be less obvious, or the moving dot needed to move faster).

Further complicating the search for suitable tasks was the need to have a **continuous measure** of task performance. The measure needed to be sensitive enough to detect potentially subtle variations in performance. A "continuous measure" was defined as a metric that could be recorded multiple times per second at a relatively constant rate. For example, events should occur approximately once every 200-400 ms (see Table 2.1), as we hypothesized that distraction effects may potentially only occur over a range of 20-300 ms. Such fine-grained metrics make it possible to detect minuscule fluctuations in performance that may have been otherwise missed.

As discussed later in the Chapter 7 and 8, finding tasks that satisfied all these criteria/constraints was a very challenging exercise. Appendix A documents some of the primary tasks that we tried and rejected, and the issues encountered.

6.4 Metrics and Measures

This section provides a high-level overview of the types of measures and metrics that could be used for analysing and comparing the noticeability and distraction of highlighting techniques. Given the objectives of our method and the availability of suitable equipment, we identified the following types of metrics and measures which could be used in our studies:

1. **Task Performance** – These are objective measures obtained by logging aspects of participant behaviour during the experiment. Examples include "Task Time", and "Distance Error".
2. **"Noticeability" and "Distraction" Metrics** – These are metrics derived from the task performance data.
3. **Eye Movements and Fixations** – By tracking the eye movements and fixations of participants using the department's Tobii TX300 eyetracker [5], we could analyse where visual attention was directed.
4. **Subjective Experience** – Subjective experience measures can reveal information about participant's thoughts, beliefs, feelings, and other effects that are harder to measuring using performance-based metrics.

6.4.1 Task Performance Metrics

Task Performance metrics are specific to the tasks performed in each study. For details about the specific task performance metrics used, consult relevant sections in the following chapters.

Overall, in both studies, the tasks were performed using the mouse. From mouse-based data, there are two main types of metric that can be used:

- **Time-based metrics** – These measure how long it took the participant to perform some task or action.

- **Mouse-movements** – These measure the characteristics of the paths traced by mouse as the participant performed the task, providing a continuous measure of task performance.

6.4.2 Noticeability and Distraction Metrics

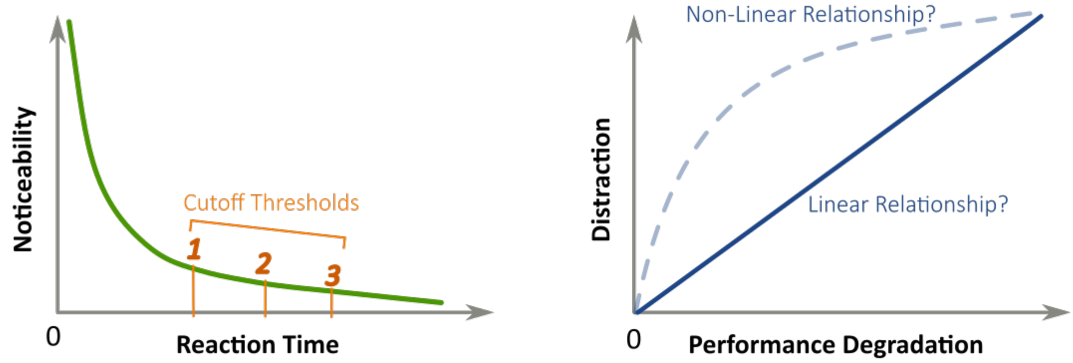


Figure 6.1: *Example graphs showing the expected relationships between: (a) Noticeability and Reaction Time, (b) Distraction and Performance Degradation. Noticeability and Reaction Time have an inverse-proportional relationship, while it is currently unknown what relationship exists between Distraction and Task Performance.*

One of the main hypotheses of this thesis (**H1.1**) is that the relative quality of highlighting techniques can be measured in terms of their Noticeability and Distraction. In our methodology, Noticeability and Distraction metrics are derived (i.e. computed) from the Task Performance Metrics, as follows:

Definition 5
*The **Noticeability Metric** (N) is defined as the proportion of time-for-noticing that was not spent searching for the highlight. This formulation is used as Noticeability (N) decreases when Reaction Time (n) increases (as shown in Figure 6.1a). Thus, when Reaction Time is low, noticeability is high (i.e. most of the time-for-noticing was not used searching for the highlight).*

$$N = (N_{cutoff} - n)/N_{cutoff} \tag{6.1}$$

N_{cutoff} is the "Cutoff Threshold" for Noticeability (i.e. the Reaction Time above which a highlight is no longer considered acceptably "noticeable" – see Figure 6.1a). These are used as the inverse-proportional relationship means that Noticeability tends to zero as Reaction Time increases. Strict thresholds (e.g. Threshold 1) use a lower maximum Reaction Time, while looser thresholds (e.g. Threshold 3) use a higher maximum Reaction Time.

In Equation 6.1, N_{cutoff} can be defined as either the "*Time to find the target item when no targets are present*" (i.e. $TaskTime_{Control}$, as used in Study 1) or "*Total time allowed for each trial*" (i.e. $TrialLen$, as used in Study 2). $TaskTime_{control}$ is preferable as it takes into account variations in participant performance. However, it can only be used if there is a target-cueing mechanism that is used across all conditions. For instance, in Study 1, the target item was indicated using slightly rounded corners, but no such mechanism was present for Study 2, hence $TaskTime_{control}$ was only used in the former.

Definition 6
*The **Distraction Metric** (D) is defined as the proportion by which task performance when distracted (i.e. d − C) was worse than in the control/baseline condition (C). It is assumed that performance degradation results in task performance that is worse (i.e. higher values) when distracted (as shown in Figure 6.1b).*

$$D = (d - C)/C \tag{6.2}$$

In Equation 6.2, d and C are measures of task performance (e.g. *Task Time* in Study 1, or *Displacement* in Study 2): d is the task performance when the highlight functions as a distraction (i.e. when the highlight does not help the notice the target), and C is the task performance in control conditions (i.e. conditions where no highlights are present).

6.4.3 Eye Tracking Metrics

Eye tracking data was used to investigate what participants were focussing as they performed the experiment. Of particular interest was questions such as:

- Where were participants looking?
- What was the first item that the participant fixated on when the stimuli appeared? Was it the target, a distractor, or something else? How long after the stimuli appeared did this happen?
- Did the participant spend a lot of time hunting for the target?
- What effect did distractors have on participant behaviour?

The eye tracking data could be processed in many different ways to obtain several types of metrics such as:

1. Time to First Fixations on X item
2. Density/Distribution of Fixations in a region
3. Frequency/Counts of Events Per Trial
4. The length of the path traced out by the fixations in a trial

6.4.4 Subjective Experience

The measures described above concern performance issues. Subjective measures or how the user feels about highlighting are also important. Candidates include asking participants about the perceived ease of identification, level of distraction, "annoyance" factor, and what impact (if any) it had on their task performance. However, it is nearly always preferable that the primary measures used are able to be objectively measured and collected. Different participants will often report their level of response to each of these measures in inconsistent ways. For example, some participants may be more willing to use extreme scores, whereas others are more conservative (i.e. central tendency bias) [83]. This causes some additional challenges when trying to analyse results from such measures.

Since survey-based measures are relatively easy to obtain, these are clearly key measures to include because there are some effects which will not be apparent if only considering performance measures. An example of this is Gluck et al.'s paper, "Matching Attentional Draw with Utility in Interruption" [74]. They found that although *there was no difference in user performance* between the three conditions trialled (when manipulating the strength of the highlighting technique used relative to the importance of the information it was drawing attention to), *there were significant differences in terms of subjective satisfaction* with the techniques (i.e. how "annoying" participants thought that each technique was).

Biometrics are an alternative way to collect subjective measures of user behaviour/performance. Examples of these include using Galvanic Skin Response [168], and various brain activity monitoring techniques such as magnetic resonance imaging (MRI's) [12] or electroencephalography (EEC) headsets [164]. However, it may not always be practical to use these measures, due to resource limitations.

6.5 Summary of Our Methodology

In this chapter, we have outlined the key principles of our experiment methodology for measuring the Noticeability and Distraction of highlighting techniques.

The following two chapters describe how these principles were employed to develop two experiment methods for measuring the Noticeability and Distraction of highlighting techniques.

7 Study 1 – Common Highlighting Techniques

7.1 Introduction

The previous chapter presented some generic principles for developing an experiment methodology for measuring the noticeability and distraction of highlighting techniques. This chapter presents a study we ran to validate these principles and to investigate the key hypotheses of this thesis (as stated in Section 1.2.3).

The primary goal of this study was to demonstrate the possibility to empirically and objectively rank highlighting techniques in terms of Noticeability and Distraction measures derived from performance-based metrics (i.e. **H1.1**). A secondary goal was to develop and validate a experiment method that directly measured the "pure" noticeability and distraction effects of the highlighting techniques being studied; in contrast, the measures of Noticeability and Distraction in many Dual-Attention studies had confounding factors such as the level of task engagement in the "primary" task. Thus, our study used an abstract visual search task as the context for measuring the effects of highlighting techniques.

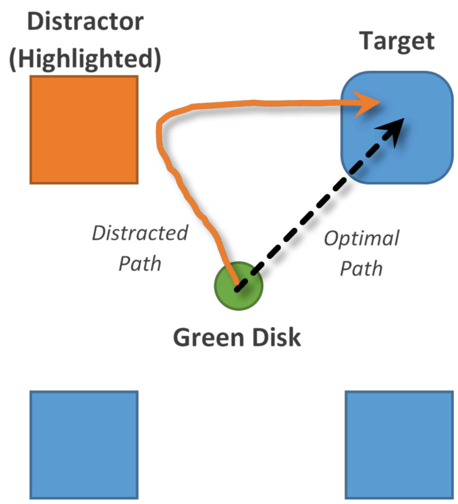


Figure 7.1: *Illustration of how our modified version of Moher et al.'s experiment worked. The participant drags the Green Disk (middle) towards the Target (top right). In theory, the disk would move along a path resembling either the Optimal Path (dashed line) or the Distracted Path (curved line), depending how distracting the highlighting effect applied to the distractor (highlighted square, top left) was.*

In service of these objectives, we first developed a new experiment method (described in more detail in Section 7.2) inspired on the work of Moher et al. [121] and Gallivan and Chapman [72], and then validated this method by using our design framework (Chapter 4) to identify four representative highlighting techniques – (i.e. `Colour`, `Pulse`, `Shake`, and `Shooting Star`) – at two levels of intensity. These techniques were applied to 1 or 2 randomly chosen items (depending on the stimulus condition) in grids containing 16 or 64 items (see Figure 7.2). During each trial, participants were required to drag a green disk (shown in the center of the grid, see Figure 7.1) on to the target item as soon as they had found it. It is important to note that the target item was the one with *slightly*-rounded corners (see Figures 7.2 and 7.8).

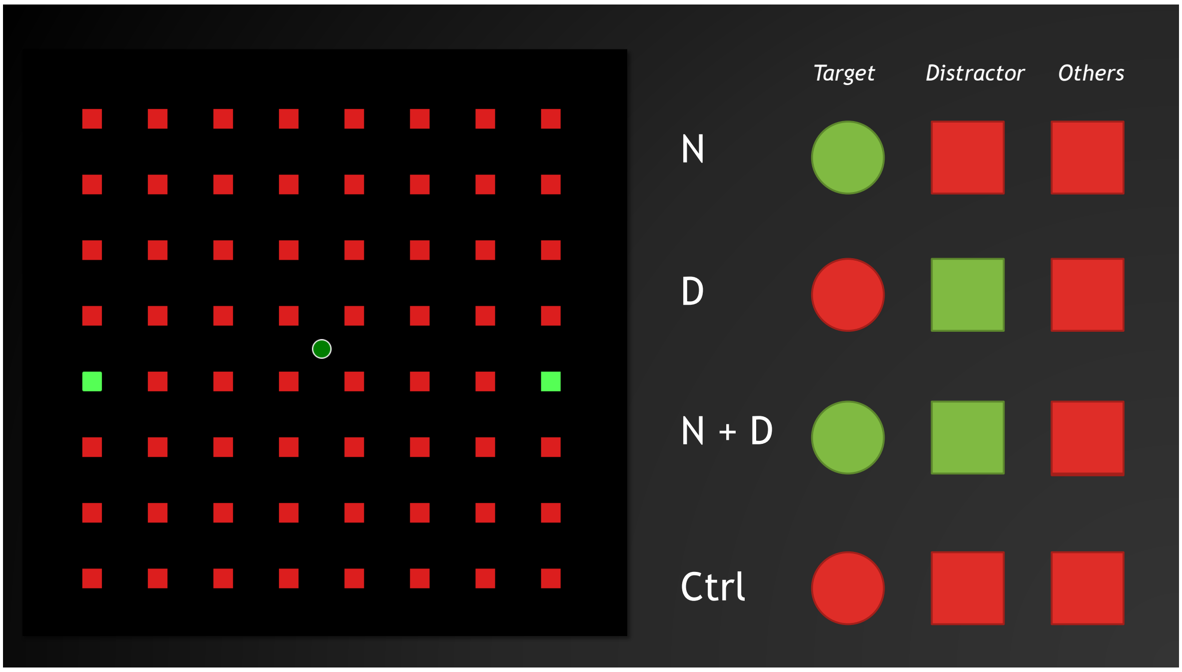


Figure 7.2: *Overview of the method for this experiment. The left hand side shows the square-shaped grid pattern of candidate items (showing the* `HColor`*-ND condition; the target is the green square on the left-side of the field). The right hand side shows the different Highlighting Patterns (i.e. which item or items in the grid were highlighted).*

Second, we validated this method by performing a user study with 20 participants (Section 7.3. The results of this study confirmed the main hypotheses of this thesis (**H1.1** and **H1.2**).

7.2 Method

This section presents the details of the design and implemention of this study. Figure 7.2 shows an overview of the stimuli used in our method. As shown on the left-hand side of Figure 7.2, participants were presented with a field of candidate items arranged in a square-shaped grid pattern. One of these items had rounded corners to indicate that it was the target; all other items had sharp corners. Highlighting effects were selectively applied to the target itself (in addition to the rounded corners), another item in the grid (with certain limitations to be discussed later), or both items at the same time. In each trial, participants needed to drag a circular green disk (located in the middle of the screen) to the target item.

As shown on the right-hand side of Figure 7.2, items were highlighted in one of three ways (or four, including the *Control* condition, where nothing was highlighted): 1) the target was highlighted (N), 2) a distractor item was highlighted (D), and 3) the target and a distractor item were both highlighted ($N + D$, or ND). The first pattern was used to measure how noticeable the highlight was, while the second was used to measure how distracting it was. The third pattern (ND) was to investigate the effects that using more than a single instance of the same highlighting technique would have (for instance, to investigate whether the effects of having two instances of the technique competing with each other). These patterns were used, as one objective of this study was to be able to measure noticeability and distraction in the same experiment session without requiring participants to perform different types of tasks or in separate passes.

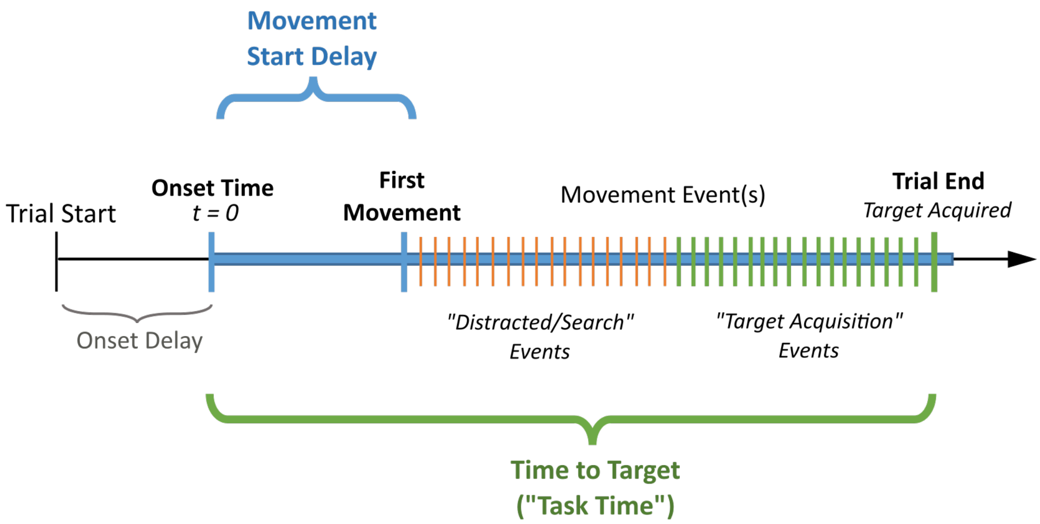


Figure 7.3: *Sequence of events during each trial. The shaded region indicates when the grid of candidate items was visible.*

Figure 7.3 shows the idealised sequence of events that occurs during each trial. The two main time-based measures of task performance were:

1. **Task Time** (or "Time to Target", TT) – Task Time represents the total time spent performing the task of identifying and acquiring the correct target item from the field of candidate items. It is an important measure of overall task performance, and can be viewed as a combined indicator of noticeability and distraction. For instance, if the target is difficult to identify, task time would be higher than if the target was easy to identify, as time consuming visual search would have been needed.

   Distraction can also influence task time: if the participant's attention was redirected towards a distractor item, or if the participant identified the wrong item as the target, the Task Time would be affected as time would have been spent travelling towards the wrong item and/or recovering from that mistake.

2. **Movement Start Delay** (MSD) – Another important time-based measure of noticeability is how long it takes for the participant to notice the target, especially when the target

is highlighted. In this experiment, the participant's reaction time should correspond to the MSD (or how long it took for the participant to begin moving towards a target after the grid of items became visible). In theory, more noticeable items should be able to attract the participant's attention quicker; thus, a shorter MSD would suggest that the highlighting techniques present helped make it easier for the participant to identify the target, while a longer MSD indicates that the participant may have had difficulty determining which item was the target. For example, the participant may not be able to find the target ,or they may have been confused by some of the distractors.

7.2.1 Highlighting Techniques

Four types of highlighting effects at two levels of intensity were used in this experiment (see Figure 7.4). These techniques were:

- **Colour** – The highlighted item was shown in a different and static colour (i.e. pink/green)
- **Pulse** – The size of the highlighted item was animated to grow and shrink several times a second
- **Shake** – The highlighted item moved side to side several times a second
- **Shooting Star** – Round particles are emitted from the cursor, and travel in a straight line towards the highlighted item

Pulse and Shake are examples of highlighting techniques using transform effects which are animated (i.e. kineticons), while Colour uses a static (non-animated) manipulation of the pixel-content of the highlighted item's visual elements. All three of these techniques are examples of techniques that are currently being used in user interfaces.

Shooting Star is a highlighting technique inspired by a number of insights/ideas from the literature. It used an extra visual element (i.e. the moving particle), as some studies suggested that it was the sudden appearance of an item which could capture the user's attention [160]. The travelling motion for this particle from the cursor to the highlighted item was used to try and draw the user's eye from their current focal point (i.e. the cursor, which should be in the center of the screen) towards the target (in peripheral vision), with the travelling motion gently encouraging the eye to smoothly track its movements.

As these techniques do not all use the same parameters, we opted for a simpler experimental design by only using "high" and "low" levels of highlighting intensity for each. The parameters values used for each condition are shown in Table 7.1.

For the health and safety of participants, it is important to note that the 6 Hz conditions (i.e. `HPulse` and `HShake`) are a potential safety hazard. This is because they exceed the recommended maximum threshold of 3 Hz [2] for preventing epileptic seizures. As a result, precautions need to be taken when recruiting participants, as well as for any medical emergencies which may occur during the experiment sessions.

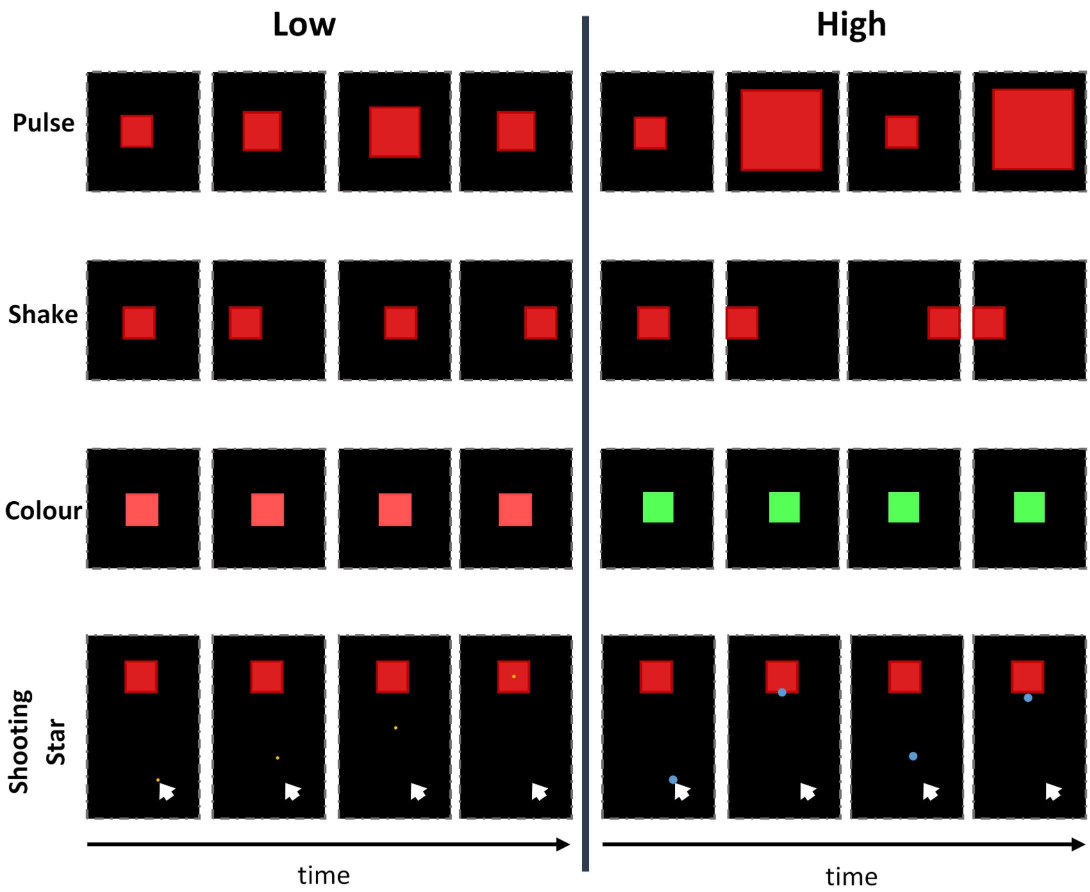


Figure 7.4: Demonstration of how the highlighting techniques behaved over time. Not drawn to scale.

Highlight Type	Low	High
Colour	Highlighted Colour = #FF5555 (Same Hue, but Brighter)	Highlighted Colour = #55FF55 (Different Hue, Brighter)
Pulse	Frequency = 2 Hz Max Scale = 1.2	Frequency = 6 Hz Max Scale = 1.5
Shake	Frequency = 2 Hz Amplitude = 5 px	Frequency = 6 Hz Amplitude = 10 px
Shooting Star	Colour = #EEAA33 Dot Size = 5 px Travel Rate = 2.5	Colour = #88DDFF Dot Size = 15 px Travel Rate = 0.5

Table 7.1: Summary of the conditions that were performed, and the levels of each factor used in each for each of those cells

7.2.2 Hypotheses

The main objectives of this experiment were to validate the main hypotheses of this thesis:

- **H1.1**: *Noticeability and Distraction Can Be Objectively Measured* – That is, the relative quality of highlighting techniques can be analysed through measurement of their emergent Noticeability and Distraction.

- **H1.2**: *More Noticeable But Less Distracting Highlighting Techniques Exist* – That is, there exists a pair of highlighting techniques, H_x and H_y, such that H_x is more noticeable and less distracting that H_y (i.e. H_x is "superior" in general).

7.2.2.1 Study-Specific Hypotheses

In addition to the overall hypotheses, we had the following secondary hypotheses:

H 7.1
Rankings derived from noticeability and distraction analysis directly correspond with subjective experience responses.

H 7.2
In $N + D$ conditions where both the target and a distractor item are highlighted, task time for $N + D$ conditions (TT_{ND}) should be greater than the task time for noticeability conditions (TT_N), and should be lower than the task time for distraction conditions (TT_D). That is,

$$TT_N \leq TT_{ND} \leq TT_D$$

H 7.3
In distraction conditions (i.e. only the distractor item is highlighted), participants will look at the distractor before looking at the target.

7.2.2.2 Highlighting Technique Hypotheses

We had the following hypotheses about the highlighting techniques included in this study, derived from prior findings in the literature (as discussed earlier in Chapter 5):

H 7.4
"Higher strength" highlighting techniques are always more noticeable and more distracting than "lower strength" techniques

H 7.5
`HColor` *and* `LColor` *are less noticeable and less distracting than the motion-based highlighting techniques*

H 7.6
`HPulse` *is the most noticeable and most distracting technique, due to the combination of rapidly repeating animation (at 6 Hz) and the distracting nature of the pulsing movement.*

H 7.7
`LColor` *is the least noticeable technique (as the colour contrast between highlighted items and non-highlighted is very low). It should also be the least distracting (as the highlights are hard to detect, and hence unlikely to affect task perforance).*

H 7.8
`HShootingStar` *should be one of the most noticeable techniques in the* $N = 64, Distance = 3$ *cases (i.e. the target is far away from the initial focal point, when a large number of candidate items are present). It should have the advantage of being able to draw the user's attention towards the highlighted item quickly, as the repeating movements always start from the user's current focal point.*

7.2.2.3 Manipulation Checks

If the method is working correctly, we should be able to make the following observations:

MC1: When the target is highlighted, participants should be able to identify it faster than in the control condition.

MC2: When distracted, participants should be expected to take longer to detect the target than when they are not distracted. So, in distraction conditions, the task time should be higher than for control conditions.

MC3: Movement Start Delay times should be less than Task Times.

7.2.3 Design

A $4 \times 2 \times 3 \times 2 \times 2$ within-subjects design was used with the following factors:

$$\begin{aligned}
\text{Highlight Type} &\in \{\text{Colour } (C), \text{ Pulse } (P), \text{ Shake } (S), \text{ Shooting Star } (SS)\} \\
\text{Highlight Strength} &\in \{\text{Low, High}\} \\
\text{Pattern} &\in \{\text{Notice } (N), \text{ Distract } (D), \text{ Notice + Distract } (ND)\} \\
\text{Grid Size} &\in \{16, 64\} \\
\text{Distance} &\in \{\text{r1, r3}\}
\end{aligned}$$

There was also a `Control` condition (i.e. "No Highlighting") that served as a baseline measurement for benchmarking the performance of the highlighting conditions. This was run with the following factors:

$$\begin{aligned} \text{Grid Size} &\in \{16, 64\} \\ \text{Distance} &\in \{\text{r1}, \text{r3}\} \end{aligned}$$

The *Highlight Strength (HS)* and *Pattern* factors were not considered the Control condition, as all levels of *HS* and *Pattern* are identical when no highlights are present.

In both the highlighting and control conditions, all the factors were fully crossed (where possible). The only exception to this was the *Distance* factor: `r3` was only available in the 64-item grid (see Figure 7.6), as the 16-item grid only had two rings (i.e. `r0` and `r1`).

Table 7.2 shows a summary of what conditions were run, with which factors fully-crossed together, and what levels of those factors were used. There were initially 153 conditions (not including the 10 training tasks).

Grid Size	Distance	HL Type	HL Strength	Pattern	Repetitions	Total
64	$\{r1, r3\}$	$\{C, P, S, SS\}$	$\{Low, High\}$	$\{N, D, ND\}$	2	96
16	$\{r1\}$	$\{C, P, S, SS\}$	$\{Low, High\}$	$\{N, D, ND\}$	2	48
64	$\{r1, r3\}$	N/A	N/A	*Control*	3	6
16	$\{r1\}$	N/A	N/A	*Control*	3	3

Table 7.2: *Summary of the conditions that were performed, and the levels of each factor used in each for each of those cells*

7.2.4 Apparatus

Stimuli were displayed on a 23 inch TFT monitor (60 Hz refresh rate, 5 ms response time, white luminance 300 cd/m2 (rated)) running at 1920 x 1080 (HD) resolution, and fitted with a Tobii TX300 eye tracking device. The experiment was run on a Windows 7 workstation with an i7-3770 processor at 3.40GHz, 8 GB RAM, and Nvidia GeForce GTX 650. Experiments were conducted in a room lit primarily using standard office lighting (fluorescent tube/strip lighting overhead), and with natural light from a nearby window (filtered by fully closing blinds). Participants performed the experiment using a wireless "Microsoft Explorer 1362" mouse (all handling characteristics were left at the operating system defaults), originating in the center of the desk space in front of the participant and free from any other physical obstructions.

The experiment software was constructed using Python 2.7.5 (64-bit) and PyQt4 (version 4.10.1 (64-bit), using Qt 4.8.5). The stimuli were implemented and rendered using the QML / QtQuick engine from the Qt framework [53], as it reduced the complexity of implementing the necessary animation effects. Stimuli were displayed in a full-screen (i.e. "maximised") window, with the titlebar and taskbar from the operating system still visible.

Eye tracking and eye tracker calibration were performed using Tobii Studio 3.3.1. Participants were told to sit comfortably such that they could clearly see the screen and freely move their mouse-arm. To optimise tracking accuracy, participants were seated approximately 60cm away from the eye tracker unit. According to the Tobii SDK User Manual [5], eye tracking accuracy is highest at this distance. No physical restraints (e.g. chin rests) were employed to ensure that the participant's head position stayed constant; instead, participants were simply instructed to sit still. According to Duchowski [61] and the Tobii SDK User Manual [5], physical restraints do not need to be used when using the Tobii Eye Tracker. During the experiment, the "Track Status" window was displayed on a secondary monitor so that we could prompt participants to adjust their posture if tracking was lost.

7.2.5 Participants

Twenty participants (13 male, 7 female; aged 18-50 with a median of 24 years) were recruited from a local university. They were all volunteers and were given a $10 voucher for a campus café for taking part in the experiment. Most were students studying Computer Science.

Participants had normal or corrected-to-normal eyesight. They were asked to not participate if they were aware of any uncorrected visual problems (e.g. colour deficiency). There were no significant eye tracking problems from participants wearing glasses (even with quite strong prescriptions).

For health and safety reasons, participants were warned that rapidly flashing and flickering graphics would be used during the experiment. Repeated efforts were made to ensure that no participants had ever suffered from epileptic seizures (particularly those related to visual stimuli) as part of the recruitment materials and again during the pre-experiment administrative processes.

7.2.6 Experiment Task

In each trial, participants were presented with a grid 16 or 64 of square-shaped items (as shown in Figure 7.5). The grid of items was shown in the middle of the screen, against a black background (`#000000`). Items were coloured "red" (`#DD2222`) with no borders. These colours were chosen to match the appearance of the diagrams/descriptions of the experiment setup in Moher et al.'s [121] paper.

All items in the grid were 32x32 px squares. The item size was chosen as icons are typically 24x24 or 48x48. Targets were spaced 75 px apart (center-to-center distance), to ensure that all items in the grid were visible on a standard HD-resolution screen when showing the 64-item condition, and that none of the items were too close to the titlebar or taskbar.

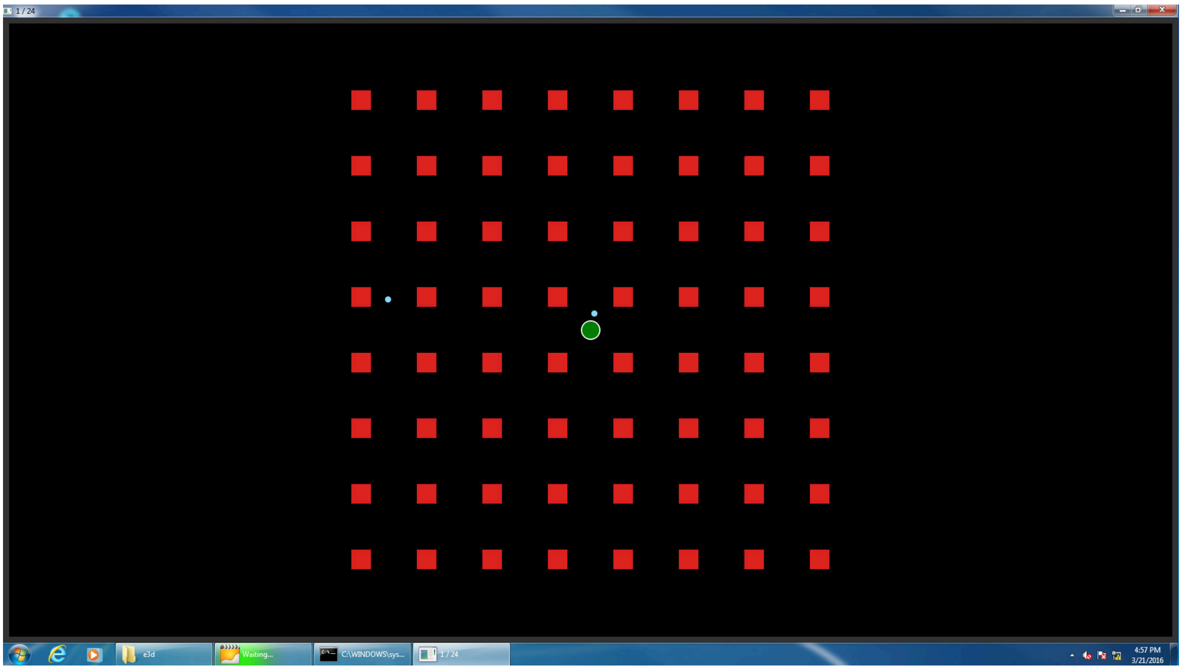


Figure 7.5: Screenshot of the experiment in progress, with a 64-item ND condition using the `HShootingStar` technique. The target in this case is item 15 (second row, second from the right)

7.2.6.1 Grid Layouts

Figure 7.6 shows how the candidate items were arranged into grids. The "N = 16" and "N = 64" grids shown here were used in the experiment. Larger grids were formed by wrapping additional "rings" of items around a smaller grid (e.g. the 64-item grid was formed by wrapping two additional rings of items – Rings 2 and 3 – around the 16-item grid). The "N = 4" grid was used in the first 5 training conditions to familiarise participants with the mechanics of the experiment.

These configurations were used so that the grid could always be divided into "quadrants". For example, in the 64-item grid, the top-left and bottom-right corners of the four quadrants were: $Q_0 = (0, 27)$, $Q_1 = (4, 31)$, $Q_2 = (32, 59)$, and $Q_3 = (36, 63)$. This design decision mattered more for the earlier versions of this experiment, where we tried to force targets and distractors to be at least 90° apart along the horizontal-axis to maximise the angle between the two (to increase the sensitivity of any distraction measurements) while still retaining control over the distance of the items from the center of the field. However, in the final experiment, this mechanism was superseeded by the "Random Onion Pie Donut" sampling scheme discussed in the following section.

For convenience and flexibility of implementation, multiple indexing schemes for referring to the grid items were created. These included the "flat array indices" (i.e. the numbers shown in each square in Figure 7.6), `(row, column)` pairs, and `(quadrant, flat-array-index)` pairs.

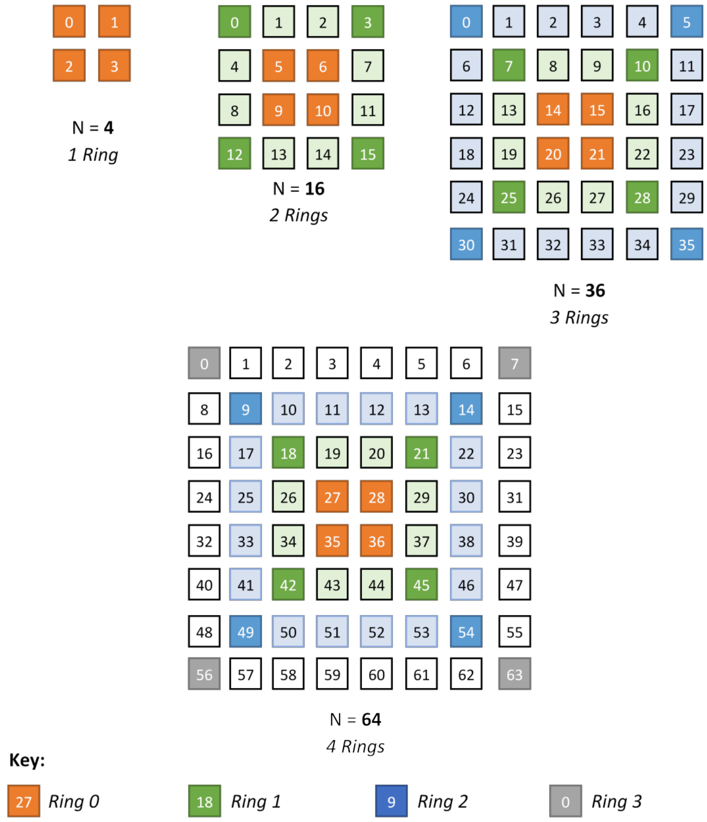


Figure 7.6: Grid configurations for candidate items. Squares with darker shading (and light coloured text) indicate the corners of each ring of items.

7.2.6.2 Target/Distractor Locations

The target and distractor items were randomly selected from the grid using a sampling scheme we refer to as the "Random Onion Pie Donut" (see Figure 7.7 for an illustration of how each item was assigned a weight for the sampling process). To select a target or distractor item, the following algorithm was used:

1. **Donut** – First, each item in the grid was assigned a score or probability weight representing the likelihood that it would be randomly selected. This was based on how many corners of the item fell within the "donut" region (illustrated as the shaded region between white rings): an item with all four corners inside the region had a score of 1.0, while an item with no corners would have 0.0. The innermost ring was at a radius of 200px from the center, and the outermost was at 400px.

2. **Pie** – Second, we included/excluded a "pie" shaped region from the set of items that the donut identified as possible candidates. Candidates were items that had a score greater than 0.1. Items with lighter/brighter shading had higher scores than those darker shading. The weights of any excluded items were changed to have negative values (indicated as the red/pink squares).

   Target – For the target, we restricted the set of candidates to only include items within the donut (i.e. lighter shaded items) which were within the target quadrant (i.e. the green squares in the top-right block of 4×4 items)

Distractor – For the distractor, we only considered the items which fell within the donut zone, did not occupy the same quadrant as the target, and were outside the collision-exclusion zone (i.e. the red "pie" slice indicated by the red lines, spaced 60° away from the line going through the center of the target). These items are indicated using blue shading.

3. **Onion** – Finally, a candidate item was randomly selected, taking into account the weights computed in the previous steps. If the *ring index* (where 0 is the innermost ring, and 3 is the outermost) matched the *Distance* required, the candidate item was used as the target; otherwise, another candidate item was selected. However, for distractor items, this "onion" constraint was skipped to make the field less predictable.

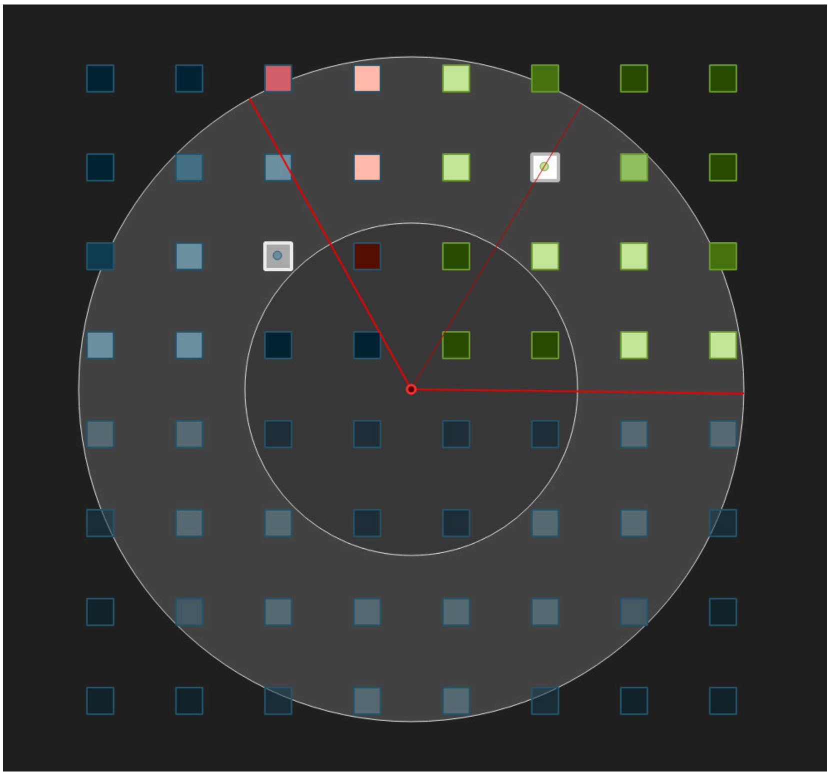


Figure 7.7: *Example of the "Random Onion Pie Donut" weighting scheme used for selecting the target and distractor items. The target is the white box with a grey border and a green dot. The distractor is the grey box with the white border and blue dot.*

This scheme was used to provide a way for having the targets be randomly selected, while maintaining some control over where the target and distractor appeared. The "donut" constraint was introduced to ensure that that the easiest targets are not used (i.e. innermost ring), and that the hardest ones (i.e. furthest away) are not used either to avoid introducing too much noise into the experiment (as visual acuity falls off with increasing distance from the center of the visual field [159] – assuming that participants were initially fixating on the central dot as instructed).

The "collision-exclusion pie" was introduced to prevent the target and distractor being too close to each other for two main reasons. First, when the target and distractor are in close proximity, there is not much angular separation between them, meaning that it becomes harder to analyse any "path deviation" effects like those that Moher et al. [121] and Gallivan and Chapman [72] identified, as it is harder to discriminate between genuine deviations induced by the distractor, and "noise" (arising from factors such as hand/desk vibrations, sensor error, and/or biomechanical limitations). Second, when the target and distractor are spaced too closely, the task also becomes too easy to perform, as simply glancing in the direction of the distractor is likely to also reveal the location of the target.

7.2.6.3 Target Cueing

As shown in Figure 7.8, targets were indicated by slightly rounding their corners ($roundness = 0.15625$, or by using $radius = 2.5px$). The roundness factor works by controlling the corner radius on the items, which is defined as:

$$radius = (size/2) \times roundness \tag{7.1}$$

where $radius$ is the corner radius and size is how wide and high the item is ($width = height = size$). A completely circular target has $roundness = 1.0$ (or $radius = size/2$), while a target with sharp corners has $roundness = 0.0$.

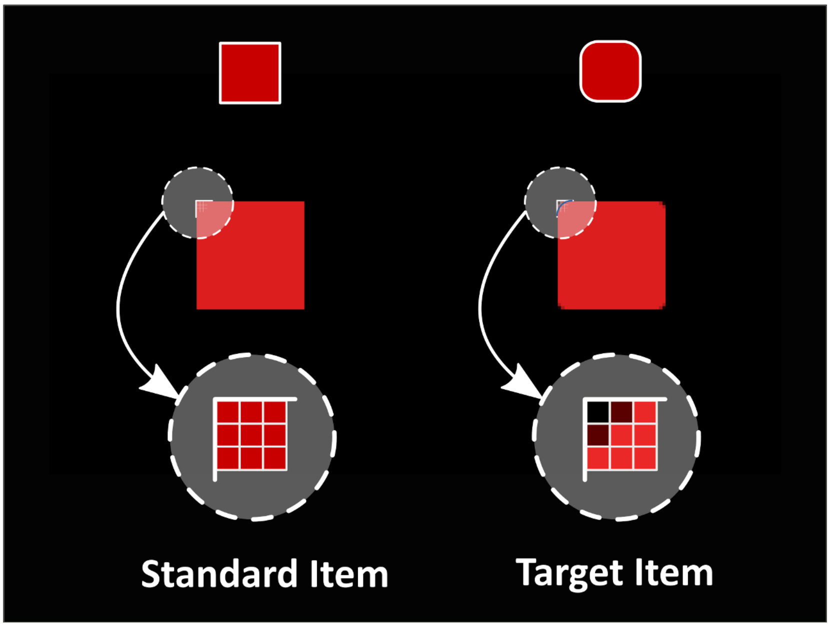


Figure 7.8: *Illustration of how the target was indicated by rounding its corners. As shown in the zoomed-in corner details, the corners of Standard and Target items differed by only 3 pixels.*

The corner rounding value used was chosen after performing several rounds of pilot testing ($N = 1$) to find a suitable value (see Appendix A.1). In order to ensure the experiment was sufficiently sensitive, the target needed to be "just noticeable" enough so that the task was not impossible, but it could also not be too obvious (i.e. it should not create a pop-out effect that is stronger than the highlighting technique being studied) so that participants would have to attempt visual search instead of being able to easily stumble across the target by chance. There was also the constraint of keeping the data collection phase of the experiment under or around 20 minutes (for a total time of 30 minutes), to avoid excessively tiring the participants.

With a radius of 2 pixels, it took an experienced participant [1] 20-25 minutes to complete the experiment; it was found that the target was nearly impossible, or impossible (the trial timed out) to locate in most trials. However, a radius of 3 pixels turned out to be "too easy" to notice, with the experiment taking 10-11 minutes to complete. Therefore, we settled on a radius of 2.5 pixels, which achieved a decent balance on time required versus difficulty of the task.

7.2.7 Experiment Procedure

The experiment had five phases: Introduction, Calibration, Training, Data Collection, and Post-Experiment Survey.

7.2.7.1 Introduction Phase

Participants were welcomed and were shown how to complete the task using a "demo" version of the apparatus. They were told that the task was to "press and hold the green disc, then drag the disc to the item with the rounded corners as fast as possible". They were also told to focus on the green disc before clicking, and to release the mouse as soon as they had positioned the disc over the target. For consistency reasons, some of the participants had to also be told to keep the mouse still until they had spotted the target, while others had to be told that they did not have to try to perfectly center the disc over the target before releasing the mouse.

The "demo" software was the same program used to conduct the actual experiment, with the only differences between the two being that the demo version had "DEMO" written in large letters across the top of the screens between trials (in place of the trial number), and that the demo version only showed a single condition. The condition used for the demos was the 4-item setup used in the first few training trials. No highlights were present during the demo (or the subsequent "Training Phase"), so participants were told that "some of the items, including the target, may or may not have some highlighting effects applied; remember that you're looking for the one with rounded corners". Originally we were planning on not making any mention of the highlighting techniques or the ways in which they would appear, but the first few participants were confused when they first encountered a trial with highlights present (i.e. N/ND/D trials) appeared.

After having the procedure demonstrated twice, participants were then asked to try to perform the task a few times to get a feel for the mouse to ensure that they correctly understood the procedure. Although a few participants only wanted to do 1-2 trials, all other participants successfully carried out 10 trials before we told them to stop.

[1]The "experienced participant" had performed similar tasks hundreds of times already; participants in the study generally took 5-10 minutes longer to complete the tasks

Participants were then reminded about the health and safety concerns (i.e. the presence of flashing and flickering graphics, and the potential for epileptic seizures resulting from this). They were asked to fill out and sign an informed consent form to confirm that they understood the procedure, its risks, and that they agreed to take part in the study.

7.2.7.2 Calibration Phase

The eye tracker was calibrated using an automated procedure provided by Tobii Studio. We made sure that participants were seated so that both eyes could be detected by the eye tracker, and that they were seated at the optimal distance (around 60cm from the screen [5]). Adjustments to posture/seating were made at this point to ensure that eye tracking was successful – this was mostly a matter of getting the participants to lean back or sit closer, but sometimes it was necessary to adjust the height of the chair (i.e. higher for shorter participants, and lower for taller participants) if a stable track could not be obtained. Participants were told to make sure that they were comfortable, as they would need to maintain this pose to hold their head still for the duration of the experiment.

Participants then performed a fully-automated calibration procedure, where they followed the movements of a large dot/disk using their eyes as it moved between 9 reference points (in a 3x3 arrangement) on the screen. They were instructed to "focus on the tiny black dot in the middle of the disk", as this reduced the need to calibrate again. This calibration procedure took less than a minute to complete. The accuracy of the calibration was checked by having the participants look at each of the reference points again, as well as at a set of points in a circle (radius 5 cm) around the center point. During this checking process, if the distance between the gaze point and the center of a target point differed by more than 7-10 mm (i.e. a "significant deviation"), the entire calibration procedure was repeated to try and improve the mapping. The calibration only needed to be repeated up to 3 times, and only for a few participants.

7.2.7.3 Training Phase

Once all the administrative and calibration steps were complete, participants began interacting with the experiment software. Before the data collection phase began, participants first completed a training phase of 10 training tasks. They were told that the data from these training tasks would not be recorded, and that these tasks would get harder after the initial few. The purpose of these training tasks was to ensure that participants had already been exposed to the 16 and 64 item configurations at least once, and that they had had some practice performing this task. This was important, as participants often expressed surprise when they first encountered these conditions (especially the 64 item condition), and had to take a few seconds to "settle down" before attempting the task for the first time.

The procedure used for these training tasks was the same as for the subsequent data generating tasks (as described in Section 7.2.7.4). Participants were shown a square-shaped grid of items, with no highlights present. Target items were cued using rounded corners (Section 7.2.6.3), just like they would appear in the data generating trials. In each trial, one of the items in the grid would be indicated as the target.

The first five trials had a 4-item grid (2x2), followed by a 16-item (4x4) trial, three 64-item (8x8) trials, and then another 16-item (4x4) trial. If any of these trials were performed incorrectly (i.e. missed target, wrong target selected, etc.) the trial would be repeated again before the end of the training phase, to ensure that all participants had correctly performed the same set of trials before beginning the data collection phase; this was particularly important in the case of the 64-item grid, as the difficulty of the task meant that it was important that we had to ensure that participants had been able to successfully find the target several times before actually performing the task.

7.2.7.4 Data Collection Phase

At the start of each trial, a circular green disk (30 pixels diameter) was shown in the center of the screen. To start the trial, participants pressed and held the left mouse button on the circular green disk. Once a trial was started, there was a short period of time (a random interval between 500-1500 ms) where the only thing visible on the screen was the circular green disk. Participants were told to fixate on the disk at this point, to ensure that it was their focal point when the stimuli first became visible. It was necessary to control where participants were looking at the start of each trial, as visual acuity falls off precipitously as distance from the focal point increases. Therefore, controlling the participant's focal point reduces the chance that particular items were favoured or penalised based on where the participant happened to be looking when the items appeared.

Following this brief onset delay, the grid of items appeared on screen. Depending on the experimental condition, None, One (N/D), or two (ND) of the items had a highlighting effect applied. To successfully complete the task, participants needed to quickly identify the target item and drag the green disk was over the center of the target. The target was indicated with rounded corners, whereas all other items had sharp corners (Figure 7.8). To end the trial, participants released the mouse button when they were ready. If the participant prematurely released the mouse (i.e. they released the mouse before reaching the target, or they released while over an incorrect target), the trial was marked as invalid, and the same task was queued up at the end of the experiment to be repeated again.

After each block of 30 trials (successful or unsuccessful), participants were presented with a screen requiring them to take a break for at least 10 seconds. There was a timer and countdown graphic to ensure that participants took this rest opportunity. Participants were told at the start of the experiment to take this opportunity to look away from the screen (e.g. at a distant target on the other side of the room), and to give their hands a rest.

At the start of each trial, the participant's progress (i.e. the current block number and trial number within the current block) was shown in large font above the disk, while instructions for how to complete the task were shown below the disk. The cursor was automatically repositioned at the start of each trial to sit exactly in the center of the disk.

7.2.7.5 Post-Experiment Survey Phase

At the completion of the data collection phase, participants were asked to complete a short survey to collect subjective experience data about each highlighting technique presented.

The survey was conducted on the same machine used for the experiment via a web-based form.

Participants were asked two questions about the highlighting techniques (see Section 7.2.8.4 for details about the questions asked). They were presented with animated images (GIF's – Graphics Interchange Format images) of each of the highlighting techniques (at both "high" and "low" levels), along with the Control Condition. These images were arranged in two columns, in a randomly chosen order (unique per participant); a small textbox beside each GIF was provided for participants to type their responses (i.e. ranking scores for each technique).

Each GIF showed two items, with one of these having rounded corners to indicate that it was the "target". For the noticeability question, the target was highlighted, and for the distraction question, the other item was highlighted. All other aspects of the presentation of the items (i.e. background colour, item colour, corner rounding, item size, and item spacing) were the same as used in the experiment. The GIF's were generated by rendering/sampling the animations at 50fps to generate 5-second long looping animations.

7.2.8 Analysis of Results

Three sets of data were collected from participants during the experiment: 1) Timing and mouse movement data, 2) eye tracking data, and 3) subjective experience responses. These datasets were then processed to compute the dependent measures and metrics reported in the results section (Section 7.3).

7.2.8.1 Time-Based Metrics

Time-based metrics were computed from the timestamps associated with events marking key points in each trial (see Figure 7.3). Two metrics were used in this study:

1. **Task Time** = Trial End – Onset Time
2. **Movement Start Delay** = First Movement – Onset Time

The data for each stimulus condition (i.e. Noticeability (N), Distraction (D), or Notice + Distraction (ND)) was analysed as three-factor within-subjects ANOVA designs (i.e. $HLType \times Strength \times GridSize$). We then repeated the analysis for each grid size, including *Distance* for the 64-item conditions (i.e. $HLType \times Strength \times Distance$). Baseline performance was derived from the `Control` condition (i.e. no highlighting applied to any items).

The data was log-transformed for the ANOVA analysis and post-hoc comparisons. This was done as these measures of time are long tailed distributions, instead of the normal distributions assumed by the statistical tests.

7.2.8.2 Noticeability and Distraction Metrics

Noticeability and Distraction metrics were calculated from the *Task Time* data using the following formulas:

$$N_T = (C - n)/C \tag{7.2}$$

$$D_T = (d - C)/C \tag{7.3}$$

where N_T is the Noticeability metric value, D_T is the Distraction metric value, n is the averaged *Task Time* for noticeability conditions, d is the averaged *Task Time* for distraction conditions, and C is the mean time for corresponding Control conditions.

Task Time was used to calculate these metrics, as this was the primary dependent measure in this experiment. All other time-based measures should all be a subset of this primary measure.

7.2.8.3 Eye Tracking Data

Eye tracking data was collected using Tobii Studio, and the raw data was exported as a CSV file for processing. The CSV file was processed to give time stamped gaze coordinates on the display. The "`LocalTimeStamp`" [2] and "`FixationPointX/Y (MSCpx)`" variables from this dataset were used in our analysis. Fixations were identified by Tobii Studio using the default "`Tobii Fixation Filter`" – the "`GazePointX/Y (ADCSpx)`" values could not be used directly due to missing data (e.g. caused by participant blinking).

The gaze data was analysed by discarding any events that did not occur during a trial, and then matching each fixation point to an item in the grid that was displayed at the time. To account for any calibration error/drift during the experiment, we applied a margin of error on each item by scaling the bounding box of each item by 175% around its midpoint. This scale factor was enough to expand the activation region for each target to reach halfway between the target and its neighbours, while reducing the number of fixations that could not be associated with an item. In some cases, it would not be possible to map the fixation to an item in the grid; in that event would then be ignored when computing any metrics requiring information about which item was fixated on. To find which item the participant was fixating on, we would check the grid cells row by row: top to bottom, left to right.

7.2.8.4 Subjective Experience Responses

Participants were asked the following two questions about the highlighting techniques (emphasis added here for clarity):

1. Please rank the techniques in order of how **noticeable** the target is, from 1 (most noticeable) to 9 (least noticeable)

[2] `LocalTimeStamp` needs to be used instead of `EyeTrackerTimestamp`. `EyeTrackerTimestamp` is a Unix timestamp (without the decimal point), obtained from the eye tracker's internal clock, whereas the `LocalTimeStamp` is a specially formatted timestamp from the computer running Tobii Studio and the experiment apparatus (i.e. host machine) clock. However, `EyeTrackerTimestamp` cannot be used as the eye tracker's clock may be out of sync with the host machine's clock, meaning that the eye tracker data cannot be easily matched back to the timing information stored in the experiment apparatus logs.

2. Please rank the techniques in terms of how **distracting** they are, from 1 (most distracting) to 9 (least distracting)

For each question, they were asked to rank the highlighting techniques from strongest to weakest in terms of noticeability (Q1) and distraction (Q2). However, for the responses to be useful as scores of relative noticeability and distraction, it was necessary to invert the order of responses so that they would go from weakest to strongest instead.

The subjective experience data was processed by first calculating the average ranking given to each highlighting technique, and then applying the following transformation to that averaged ranking to obtain a noticeability or distraction score:

$$s' = (9 - s) + 1 \tag{7.4}$$

where s is the averaged subjective response for a given highlighting technique, 9 is the highest rank that could be assigned (i.e. for the "weakest" technique), and 1 was used to adjust the values so that the Control Condition would be at $(N = 1, D = 1)$.

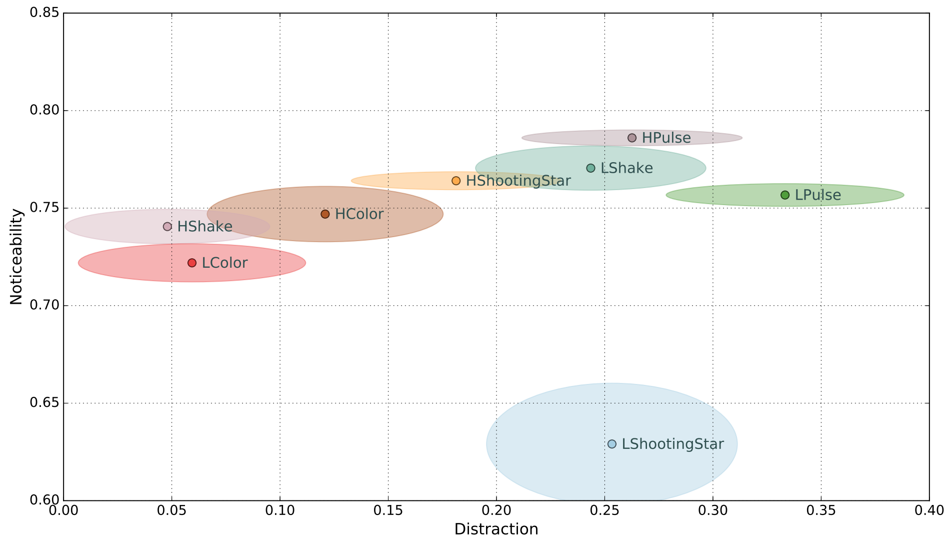
7.3 Results

7.3.1 Noticeability versus Distraction

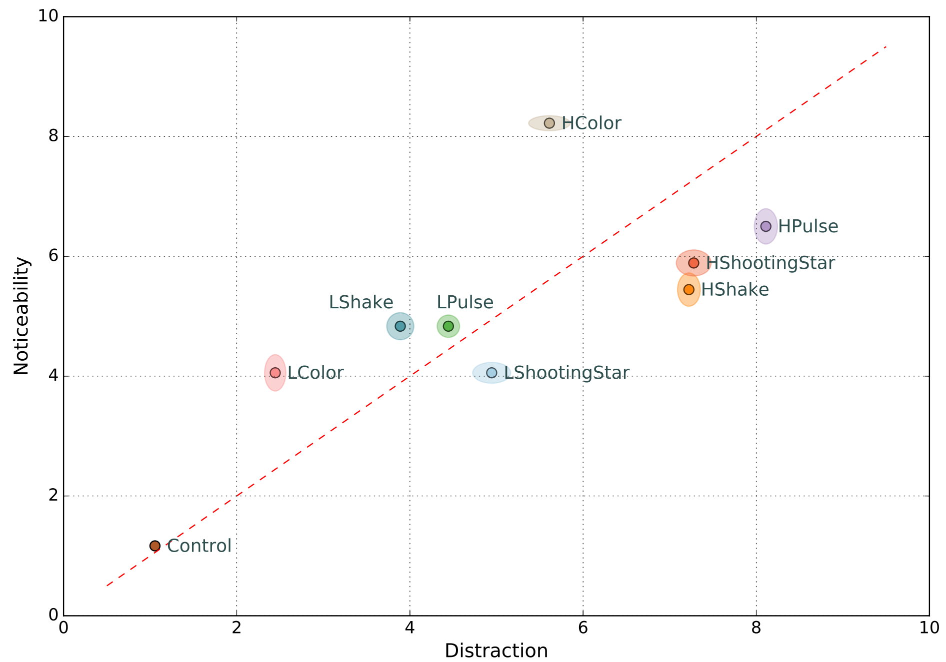
Figure 7.9 compares the relationship between Noticeability and Distraction for different highlighting techniques. Figure 7.9(a) uses metrics N_T and D_T computed from the *Task Time* data using equations 7.2 and 7.3. Figure 7.9(b) uses metrics N_S and D_S computed from the subjective experience responses using equation 7.4. In both figures, techniques closer to the top left corner are better than those in the bottom right corner (i.e. they are more noticeable and less distracting). The shaded regions around each point show ± 1 standard error. Table 7.3 shows a summary of the mean and standard error for N_T and D_T.

	Noticeability (N_T)	*N_T Error*	**Distraction (D_T)**	*D_T Error*
HPulse	0.786034	0.008100	0.262623	0.101776
LShake	0.770521	0.022700	0.243545	0.106436
HShootingStar	0.764032	0.009233	0.181341	0.096856
LPulse	0.756726	0.011647	0.333313	0.109882
HColor	0.746946	0.028449	0.120854	0.109026
HShake	0.740559	0.017690	0.047968	0.094551
LColor	0.721940	0.019541	0.059335	0.105037
LShootingStar	0.629049	0.062404	0.253345	0.115891

Table 7.3: *Mean and standard error values for the Noticeability (N_T) and Distraction (D_T) metrics for each highlighting technique*



(a) **Task Performance** - N_T and D_T - `LPulse` and `LShootingStar` are clear outliers – both represent techniques which are less noticeable but more distracting (i.e. "worse") than many of the others. Also note the differences in scale and start/end points for each axis.



(b) **Subjective Experience** - N_S and D_S - The dotted line represents the simplest case of N_S monotonically-increasing with respect to D_S, when they are exactly the same (i.e. $N_S = D_S$). Although the points do not fall on this line, there is still a strong positive correlation between N_S and D_S (Spearman $\rho = 0.8236$)

Figure 7.9: *Comparison of measures of Noticeability and Distraction, using performance-based metrics (top) and subjective experience measures (bottom). Techniques closer to the top-left corner are better. Shaded regions show* ± 1 *standard error.*

7.3.1.1 Relationship Between N_T and D_T

As predicted in **H1.2**, **there does not appear to always be a monotonically-increasing relationship between noticeability and distraction**. As shown in Figure 7.9(a), it is possible to find highlighting techniques which satisfy the two conditions:

- It is possible to find a highlighting technique which is less noticeable and more distracting (i.e. worse) than a given one (*Condition 1*). For example, `LShootingStar` ($N = 0.6290, D = 0.2533$) is the least noticeable technique overall, and is the third most distracting technique (behind `HPulse` ($D = 0.2626$) and `LPulse` ($D = 0.3333$), making it worse than all the others.
- It is also possible to find one or more highlighting techniques which are "better" (i.e. more noticeable and less distracting) than a given technique (*Condition 2*). For example, `HShootingStar` ($N = 0.7640, D = 0.1813$), `LShake` ($N = 0.7705, D = 0.2435$), and `HPulse` ($N = 0.7860, D = 0.2626$) are all more noticeable and less distracting than `LPulse` ($N = 0.7567, D = 0.3333$).

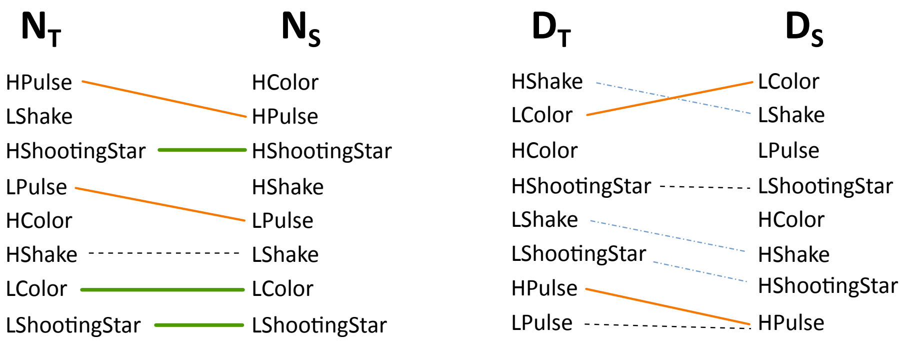
As seen in Figure 7.9(a), the majority of the techniques appear to fall along a straight line. Applying a simple linear regression to the data, there is a strong correlation between noticeability and distraction if `LShootingStar` is excluded ($\mathbf{R^2} = \mathbf{0.5614}$, $D_T = 3.8433, N_T - 2.7243$). If both `LShootingStar` and `LPulse` (i.e. the two outlier techniques) are excluded, there is an even stronger correlation ($\mathbf{R^2} = \mathbf{0.8893}$, $D_T = 3.7465, N_T - 2.676$). However, if all datapoints are included in the analysis, there is no correlation between N_T and D_T (i.e. $\mathbf{R^2} = \mathbf{0.003}$, $D_T = 0.1163N_T + 0.1018$). This is further evidence in support of **H1.2**: when `LShootingStar` and `LPulse` are included in the regression analysis results, there is a weaker correlation between N_T and D_T, as these techniques occur outside the monotonically-increasing region where all the points are all strongly correlated.

7.3.1.2 Correspondence Between Noticeability and Distraction Metrics – Task Performance versus Subjective Experience

Figure 7.10 shows a comparison of the relative quality of highlighting techniques as measured using the task performance (T) and subjective experience (S) metrics. Thick green lines indicate techniques which were ranked in the same way in both T and S. Solid lines indicate techniques whose order changed by one [3] place between T and S. Dotted lines indicate when similar techniques (i.e. same effect, different strength/intensity) have a similar rank in T and S.

Although there are some similarities in the relative noticeability between T and S, overall, we **were unable to reject the null-hypothesis that T rankings are not correlated with S ratings** (see **H 7.1**). In other words, our metrics of task performance (N_T, D_T) do not correspond with our metrics of subjective experience (N_S, D_S). Possible explanations for this result and its implications are discussed in Section 7.4.4.5.

[3] Only techniques differing by one place are indicated, since these are still "close enough" to potentially be considered the same (especially if the confidence intervals between the technique and its neighbours overlap).



(a) Comparison of how noticeable each highlighting technique is (from most to least)

(b) Comparison of how distracting each highlighting technique is (from least to most)

Figure 7.10: *Comparison of relative quality of highlighting techniques (from best to worst) in terms of noticeability and distraction, as measured from Task Performance (X_T) compared to Subjective Experience (X_S).*

7.3.1.3 Characteristics of the Highlighting Techniques

In Figure 7.9, two points stand out in particular: `HColor` in the Subjective Experience graph (Figure 7.9(b)), and `LShootingStar` in Figure 7.9(a).

1. `HColor` stands out for being seen by participants as the "best" overall (i.e. most noticeable, while only being moderately distracting). However, objectively speaking, `HColor` is probably a "second-tier" (see Figure 7.9(a)) technique in the sense that the most noticeable technique for the lowest amount of distraction is `HShake`, while the most noticeable technique (`HPulse`) does so by being more distracting than it is.

2. `LShootingStar` stands out for being the least noticeable and third most distracting technique. Although it may not be possible to rank/order 2D points, `LShootingStar` appears to be the worst of the highlighting techniques studied here, as most are more effective (noticeable and less distracting) than it is. The other candidate for the "worst" technique is `LShake`. That was ranked as being the most distracting technique, but has the benefit of being the second most noticeable technique. Therefore, `LShake` would still be more useful than `LShootingStar`, as it could be used effectively if a highly noticeable technique is required and none of the less distracting techniques can be used instead (e.g. if they are all being used already).

In Figure 7.9(b), there appear to be two distinct clusters of techniques (not counting the single points for `Control` and `HColor`): "Low Strength" techniques and "High Strength" techniques. The "Low Strength" techniques were mostly perceived as being more noticeable than they were distracting, even if their "power" was low. However, the "High Strength" techniques (or to be more precise, all the high strength techniques involving motion) were all rated as being really distracting. In general, all the techniques "below the line" were all techniques which involved motion.

In Figure 7.9(a), there are several interesting trends:

- Lower strength techniques were more distracting than the higher strength ones, except for `Color`.

- Higher strength techniques were more noticeable, except for `Shake`

7.3.1.4 Summary of Noticeability versus Distraction Findings

We can draw the following conclusions about the relationship between Noticeability and Distraction of the highlighting techniques studied:

- As predicted by **H1.2**, we found evidence that noticeability does not increase monotonically with increasing distraction. That is, it is possible to find techniques which do not follow the monotonically-increasing relationship.
- Rankings of the noticeability and distraction characteristics of highlighting techniques obtained using our performance-based metrics (N_T, D_T) do not correspond to rankings obtained via subjective experience questions
- The most *noticeable* highlighting technique was `HPulse` (High-Strength Pulse)
- The most *distracting* highlighting technique was `LPulse` (Low-Strength Pulse).
- Overall, `LShootingStar` was the least effective highlighting technique. It was the least noticeable and most distracting. `LPulse` was the second-least effective technique, as there are techniques which are more effective but less distracting.
- Although `HColor` was rated as being the "best" overall (i.e. most noticeable but only mildly distracting), it was one of the least noticeable according to the performance-based measures.

The metrics used here were calculated from the task time data. However this raises several questions, such as: Was task time really the best measure to use for this purpose, and what effects (if any) did the various factors being manipulated have on user performance with the highlighting techniques? These questions are answered in the following sections, which analyse the data for the different sets of metrics that were collected.

7.3.2 Task Time

Task Time is the primary measure of user performance in this experiment. It measures how long it took the participant to select the target item once the grid of candidate items appear. In this section, we examine the task time data to verify that it is a sensitive measure for quantifying the effects of highlighting techniques, and to understand what effects manipulating the independent measures had on user behaviour.

Figure 7.11 compares how task times vary across different combinations of highlighting techniques and stimuli conditions. It validates the following Manipulation Checks:

1. $TT_N < TT_C$ – Task times for noticeability conditions (TT_N) are substantially lower than those for control conditions (TT_C)
2. $TT_D \geq TT_C$ – Task times for distraction conditions (TT_D) are slower than those for control conditions

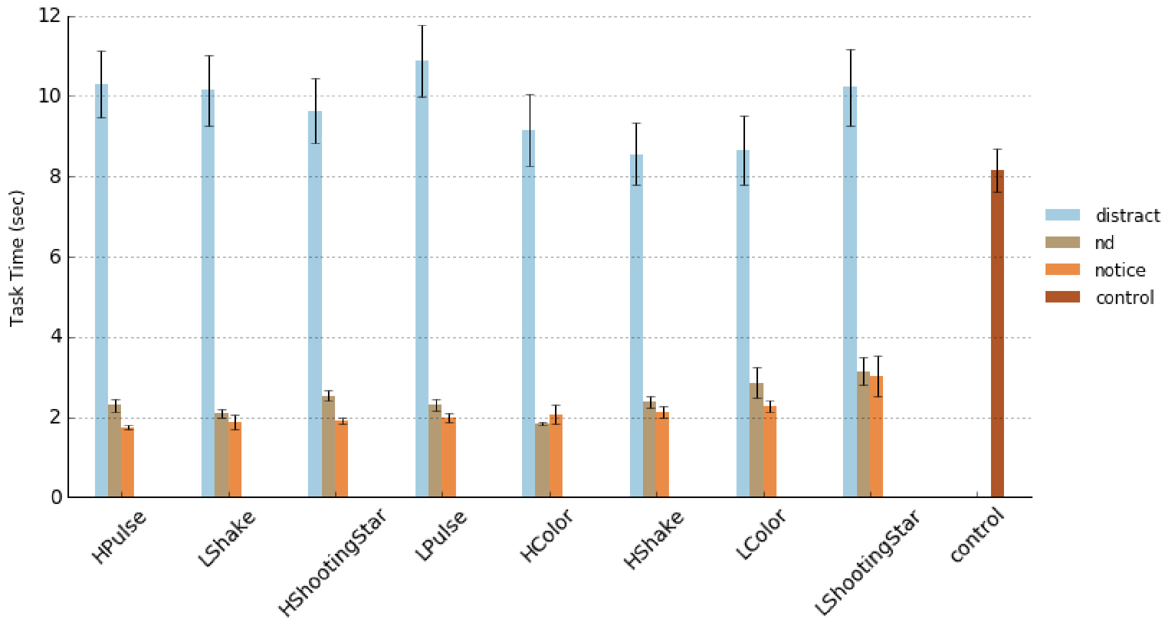


Figure 7.11: *Overview of how mean task times vary between different combinations of highlighting techniques and stimuli conditions. The techniques are shown in order of decreasing noticeability (lower is better). Note how* `LPulse` *and* `LShootingStar` *appear to be outliers in the general trends for TT_N and TT_D trials.*

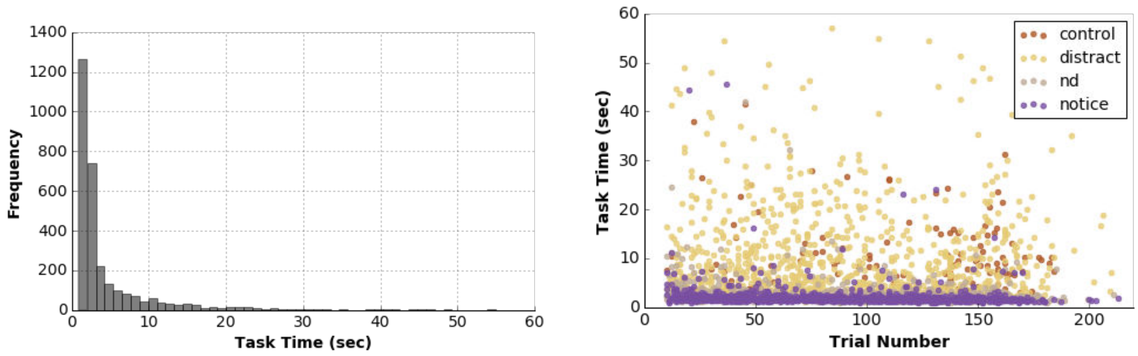
Figure 7.11 shows that task times for *ND* conditions (i.e. TT_{ND}) are more similar to task times for noticeability conditions (TT_N) than for distraction conditions (TT_D). This suggests that in the *ND* conditions, the distractors functioned less as distractors, and more as visual search aids. That is, the use of highlighting in worked to reduce the search space.

In trials where the target and a distractor item were highlighted, **TT_{ND} behaved as predicted in H7.2**. That is, $TT_N \leq TT_{ND} \leq TT_D$ was true for *most* of the highlighting techniques. `HColor` is an interesting case as the mean task time for noticeability trials was higher than in trials where both the target and distractor were highlighted ($TT_N = 2.063583(SE = 0.232)$, $TT_{ND} = 1.834417(SE = 0.048)$). However, this difference is not statistically significant, so although we cannot completely reject the null-hypothesis that TT_N is strictly less than TT_{ND}, we can at least conclude that they may at least be equal.

Another interesting finding is that there was a larger difference between TT_N and TT_{ND} for `HPulse`, `HShootingStar`, and `LColor`. This suggests that those techniques were more distracting when they are applied to multiple items at the same time (i.e. "self-distracting"), as the presence of other highlighted items made it take longer to locate the target.

7.3.2.1 Distribution of Task Times

Figure 7.12(a) shows a histogram of the task times across all conditions. It shows that *Task Time* was a long-tailed distribution, with mean = 4.936 sec (min = 0.88 sec, max = 57.17 sec). Therefore, the task time data was log-transformed before analysis to satisfy the normality assumption of the statistical tests.



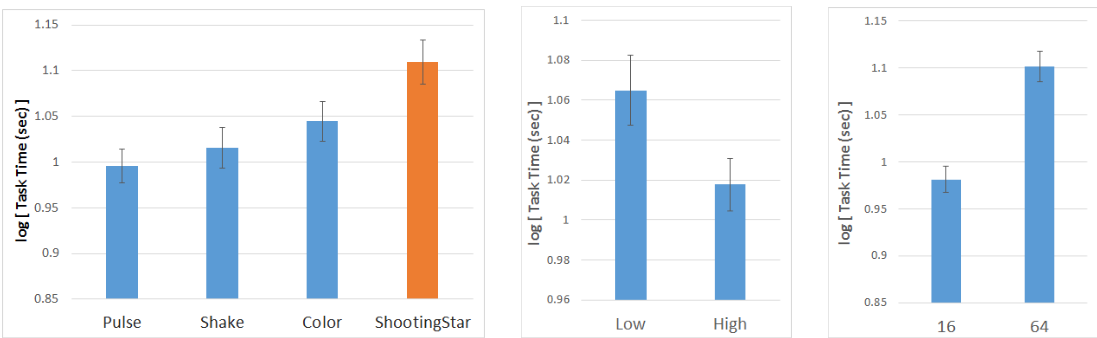
(a) Histogram of Task Time frequencies (b) By condition type

Figure 7.12: *Plots showing overall distribution of task times*

Figure 7.12(b) shows a plot of the task times based on *Pattern* (i.e. whether it was a Noticeability, Distraction, *ND*, or Control condition). Most of the Noticeability and *ND* trials had low task times (although there were a few outliers), while *Control* and *Distraction* conditions had the highest task times.

7.3.2.2 Task Times for Noticeability

We analysed the log-transformed task time data for noticeability trials using a three-factor within-subjects ANOVA. As expected, there were significant differences between the mean task times for all three factors: *Highlight Type* at the ($F_{3,57} = 13.545$, $p = 0.000$), *Highlight Strength* ($F_{1,19} = 7.785$, $p = 0.012$), and *Grid Size* at the ($F_{1,19} = 46.689$, $p = 0.000$).



(a) Highlight Type (b) Strength (c) Grid Size

Figure 7.13: *Comparison of the mean task times showing the differences between the levels of each factor for noticeability conditions. The log-transformed times are shown here to make the differences clearer to see. Lower task times are better.*

Figure 7.13 shows that *Highlight Strength* and *Grid Size* behaved as expected: "higher strength" highlighting techniques were noticed faster (Figure 7.13(b)), and targets were noticed faster when there were fewer items on screen (Figure 7.13(c)). In Figure 7.13(a), it can be seen that task times for the two other motion-based highlighting techniques (`Pulse` and `Shake`) were lower (i.e. more noticeable) than `Colour`. It can also be seen that `Shooting Star` (highlighted in orange) had the highest mean task time, and that it appears to be significantly different. Pairwise comparisons (using Tukey HSD at $alpha = .05$) confirmed that there were significant differences between `Pulse–Shooting Star`, and `Shake–Shooting Star`. However, there were no significant differences between `Pulse–Shake`, and/or between `Color` and the other techniques.

We also found significant interactions between most combinations of the factors (see Figure 7.14):

- **Highlight Type × Strength** – ($F_{3,57} = 8.007$, $p = 0.000$)
- **Strength × Grid Size** – ($F_{1,19} = 5.358$, $p = 0.032$)
- **Highlight Type × Strength × Grid Size** – ($F_{3,57} = 3.312$, $p = 0.026$)

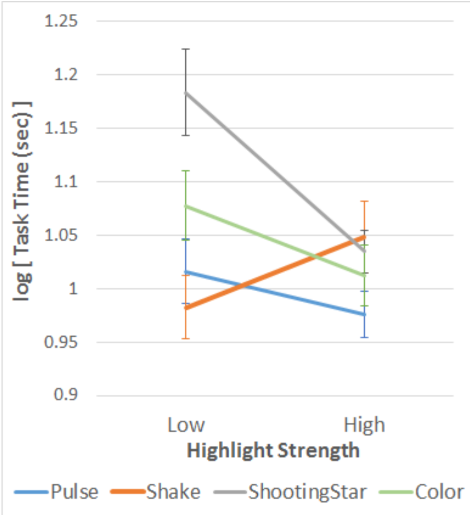
However, there was *no* significant interaction found for *Highlight Type* and *Grid Size* ($F_{3,57} = 0.245$, $p = 0.865$), suggesting that the difference between highlighting techniques may be robust to changes in grid size.

Figure 7.14(a) shows the interactions for *Highlight Type* × *Highlight Strength*. The most notable effect was that performance with `Shake` got worse when the strength of the highlighting effect was increased, whereas the opposite was true for all the other highlighting techniques. As seen in Figure 7.14(a), the *Low* strength `Shake` effect (`LShake`) had the lowest task time (among low strength techniques), while the *High* strength `Shake` effect (`HShake`) had the highest task time (among high strength techniques). This suggests that while the shake effect itself might be quite noticeable, it was difficult for participants to complete the task in `HShake` conditions. Reasons for this may be that there was an interaction between the highlighting technique and the target cueing technique making it harder to see the target, or that the shaking movement may have made it more difficult to place the cursor over the item.

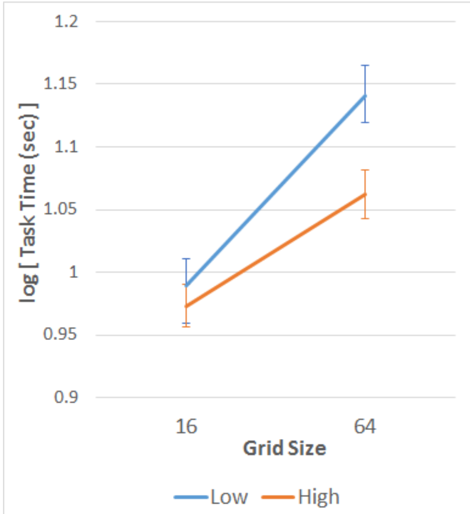
With all the other highlighting techniques in Figure 7.14(a), task time decreased (i.e. improved) when going from the *Low* strength to the *High* strength instantiations. With `Pulse` and `Colour`, the effect of the highlight strength manipulation was similar (i.e. they were nearly parallel), whereas there was a larger difference (i.e. a steeper gradient) between `LShootingStar` and `HShootingStar`.

Figure 7.14(b) shows the interactions for *Highlight Strength* × *Grid Size*. Although this is not one of the main interactions that we are interested in, it does show that performance with lower strength highlighting techniques (blue line) degraded more than higher strength highlighting techniques (orange line) when the number of candidate items increased. That is, *lower strength highlighting effects are generally less suitable for information spaces containing many items*.

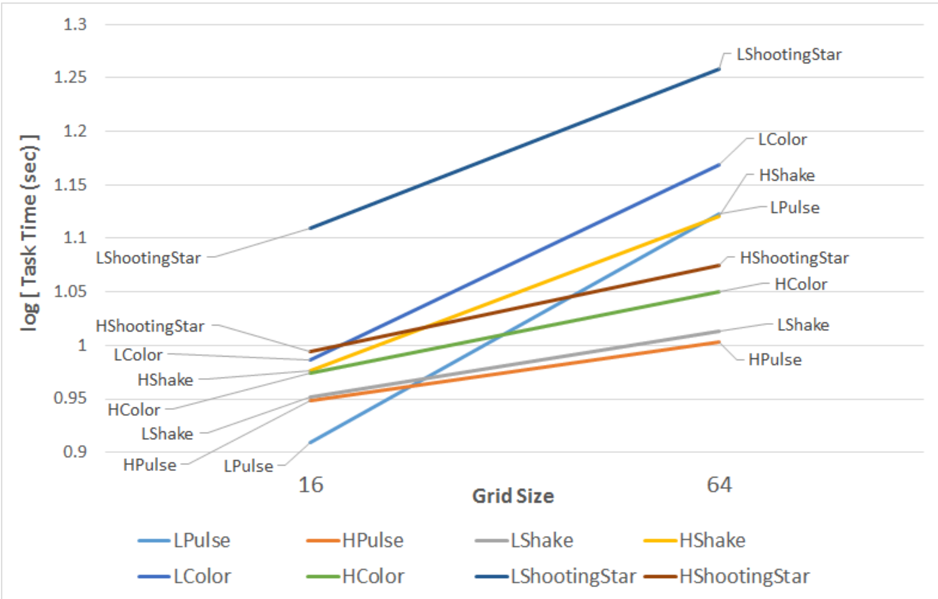
Figures 7.14(c) and 7.15 show the interactions between the three main factors. Figure 7.15 confirms that task times always increased when *Grid Size* increased; this was to be expected, since the search space is much larger. It also shows that the differences between the high and low strength instantiations of each technique were greatest when *Grid Size* was higher. For example, task times for `Shake` and `Color` were nearly the same for the low and high strength instantiations when *Grid Size* $= 16$. In contrast, there was a clear separation between `LShootingStar` and `HShootingStar` in Figure 7.15(d), with the two plots almost parallel, indicating that there were no significant interactions between *Grid Size* and `Shooting`



(a) Type × Strength



(b) Strength × Grid Size



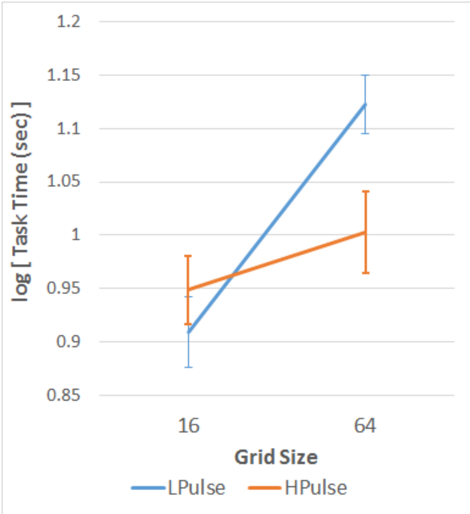
(c) (Type × Strength) × Grid Size

Figure 7.14: *Plots of the significant interactions between the factors of the noticeability conditions. The log-transformed mean task times are shown here to make the differences clearer to see. Lower task times are better.*

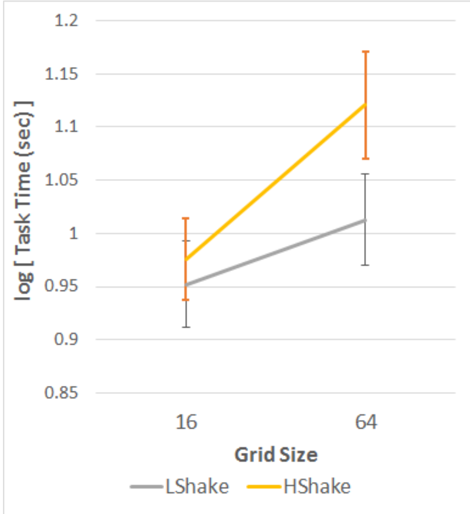
`Star`.

Figure 7.15(a) shows an interesting interaction between the `Pulse` technique and *Grid Size*: there was a crossover effect, where the log-transformed task times with `LPulse` degrade significantly when *Grid Size* increases, whereas the log-transformed task times with `HPulse` were more stable. This interaction is also visible in Figure 7.15, where `LPulse` went from being the best technique at *Grid Size* = 16 to being the third worst at *Grid Size* = 64.

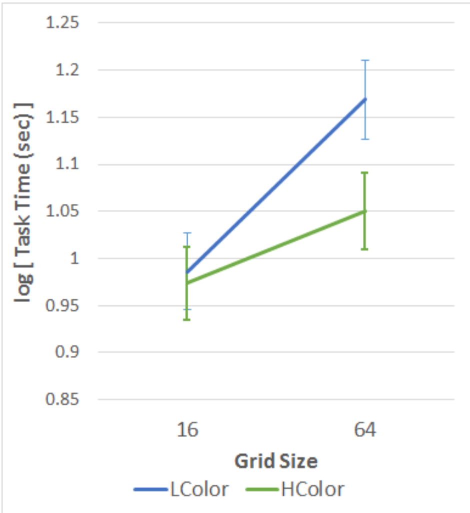
We also analysed the effects of the *Distance* factor (i.e. which ring the highlighted target was



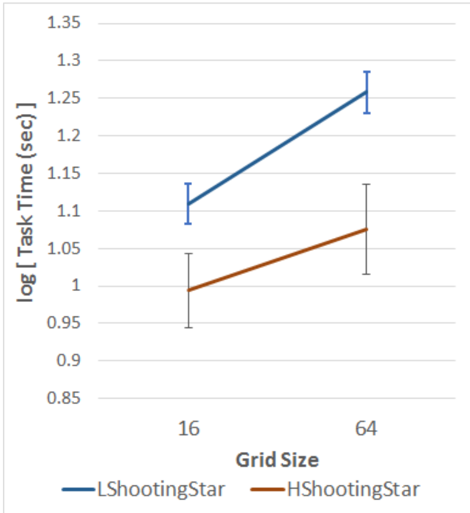
(a) Pulse



(b) Shake



(c) Color

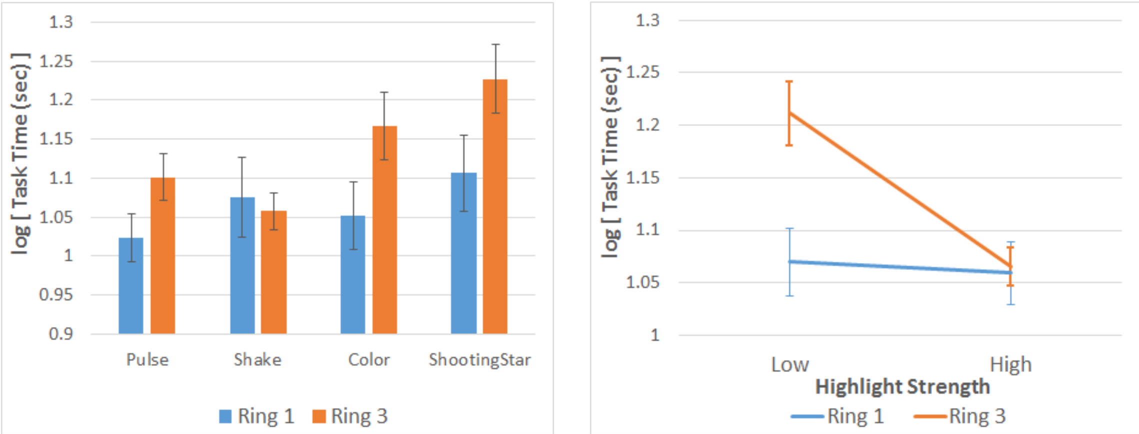


(d) Shooting Star

Figure 7.15: *Plots of the Type × Strength × Grid Size interactions – one for each highlight type – showing the relationship between the Low and High strength instantiations of each technique and Grid Size. Log-transformed mean task times are shown here to make the differences clearer to see. Lower task times are better.*

in; a higher ring index is further away from the initial focal point). Since this factor was only valid for the 64-item case, we performed a separate analysis using a three-factor ANOVA within-subjects design (*Highlight Type* × *Highlight Strength* × *Distance*) for the 64-item task time data for noticeability conditions. There were no significant effects for the *Distance* factor itself ($F_{1,19} = 3.991$, $p = 0.060$), but there were significant interactions between *Highlight Type* × *Distance* at the $p \leq 0.05$ level ($F_{3,57} = 2.768$, $p = 0.050$), and between *Highlight Strength* × *Distance* ($F_{1,19} = 8.566$, $p = 0.009$).

Figure 7.16(a) shows that performance with `Shake` improved when the distance to the target was higher, while for all other techniques, performance degraded by a consistent amount.



(a) Type × Distance (b) Strength × Distance

Figure 7.16: *Graphs showing the significant interactions between the Distance factor and other factors. The log-transformed mean task times are shown here to make the differences clearer to see. Lower task times are better.*

While the latter follows common expectations, the former is a counter-intuitive result. Figure 7.16(b) shows that it took participants longer to notice *Low* strength highlighting techniques when they were far away from the target, but when the *High* strength versions were used, distance had no discernible effect on task times. This suggests that high strength techniques were better able to capture user attention, despite our peripheral vision being less sensitive to visual differences in general.

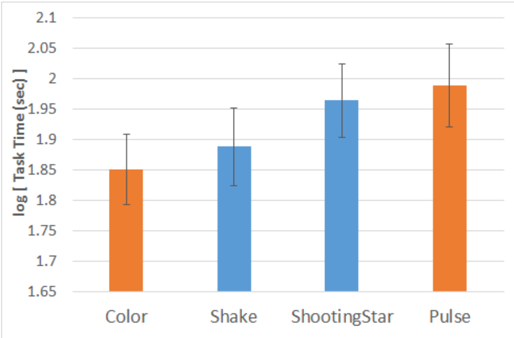
Therefore, in conclusion, these results confirm that *Task Time* is a sensitive measure of the differences between the "noticeability" effects of different highlighting techniques and the manipulations of those techniques.

7.3.2.3 Task Times for Distraction Conditions

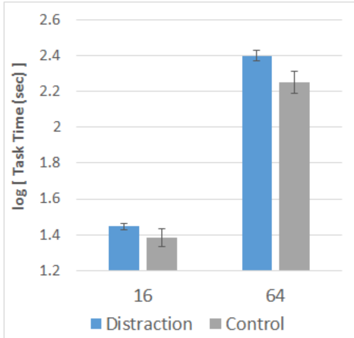
We repeated the analyses from the previous section (Section 7.3.2.2) for noticeability conditions on the task time data for distraction conditions. There were significant effects for *Highlight Type* ($F_{3,57} = 4.974$, $p = 0.004$), *Grid Size* ($F_{1,19} = 569.078$, $p = 0.000$), and a significant interaction between *Highlight Type* × *Grid Size* ($F_{3,57} = 3.548$, $p = 0.020$).

However, unlike the analyses performed for the noticeability conditions, there was no significant effect detected of *Highlight Strength* ($F_{1,19} = 2.105$, $p = 0.163$), nor any interactions involving this factor. The experiment procedure lacks sensitivity to reliably identify differences between the different highlighting strengths.

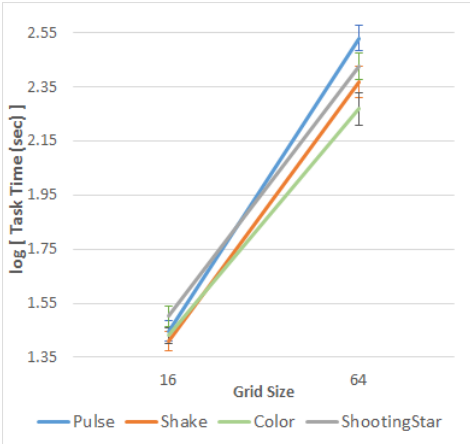
Figure 7.17(a) shows the mean log-transformed task times for different types of highlighting techniques (*Highlight Type*), sorted from least to most distracting. Pairwise comparisons using Tukey HSD tests at $alpha = 0.05$ found that the only significant difference was between `Color` and `Pulse`. Interestingly, the order here corresponds to the subjective experience ratings for the *High* strength instantiations (as shown in Figure 7.10(b)).



(a) Highlight Type



(b) Grid Size



(c) Type × Grid Size

Figure 7.17: *Comparison of the mean task times showing the differences between the levels of each factor for distraction conditions. Log-transformed times are shown here to make the differences clearer to see. Lower task times are better, as higher task times represent techniques which were more distracting. Orange shading in (a) indicates techniques where significant differences were found.*

Figure 7.17(c) shows the effect of *Grid Size* on log-transformed mean task times for distraction conditions. It also compares this effect with the effect that this factor has on `Control` conditions. It shows that most of the difference was due to the increased difficulty of the search task (as shown by the Control conditions) as opposed to being caused solely by the distraction effects of the highlighting technique. However, *Grid Size* did still appear to have an effect on distraction task times, as the difference between distraction and control conditions was greater at *Grid Size* $= 64$.

Figure 7.17(c) shows the interactions between *Highlight Type* × *Grid Size*. Overall, it can be seen that the task times for all techniques are quite similar, though there is a slightly greater spread between them at *Grid Size* $= 64$. More interesting is the presence of two pairs of crossover effects: the first between `Pulse` and `ShootingStar`, and the other between `Shake` and `Color`. In each of these crossover effects, the "better" or less distracting effects at *Grid Size* $= 16$ (i.e. `Pulse` and `Shake`), performance degraded with the larger grid size.

In conclusion, these results show that *Task Time* is a measure that can capture some of the differences between the "distraction" effects of different highlighting techniques. However, it should also be noted that these measurements are less sensitive to the differences between highlighting techniques and their manipulations than the ones for noticeability conditions, as there is a lot more noise in the data.

7.3.2.4 Task Times for ND Conditions

In the *ND* conditions, both the target and a distractor item were highlighted. As shown in Figure 7.11, task times for these conditions (TT_{ND}) were closer those for noticeability (TT_N) than those for distraction (TT_D) conditions. Therefore, in this section, we address the question of how similar TT_{ND} was to TT_N and/or TT_D.

We analysed the log-transformed TT_{ND} data using the same methods used for the noticeability and distraction task times discussed in previous sections. As shown in Table 7.4, there were significant effects for all factors, as well as all combinations of those factors.

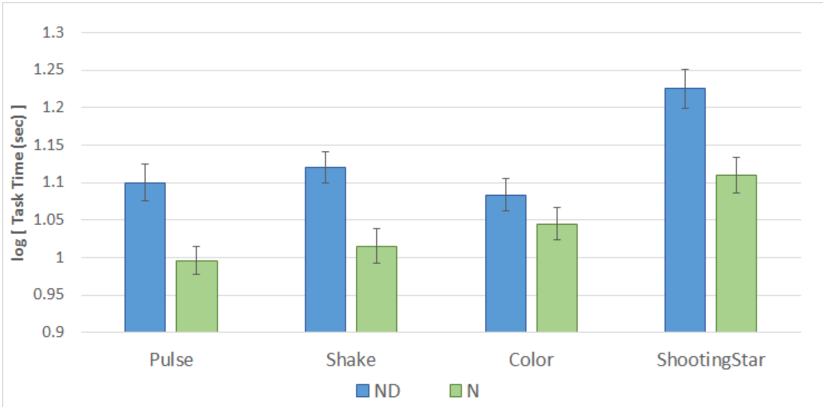
Factors	F Ratio	*p*-value
Highlight Type	$F_{3,57} = 9.387$	0.000
Highlight Strength	$F_{1,19} = 5.027$	0.037
Grid Size	$F_{1,19} = 57.586$	0.000
Type × Strength	$F_{3,57} = 7.962$	0.000
Type × Grid Size	$F_{3,57} = 3.798$	0.015
Strength × Grid Size	$F_{1,19} = 6.545$	0.019
Type × Strength × Grid Size	$F_{3,57} = 3.384$	0.024

Table 7.4: *Results of three-factor ANOVA analysis for ND task time data. There were significant effects for all factors and all combinations of those*

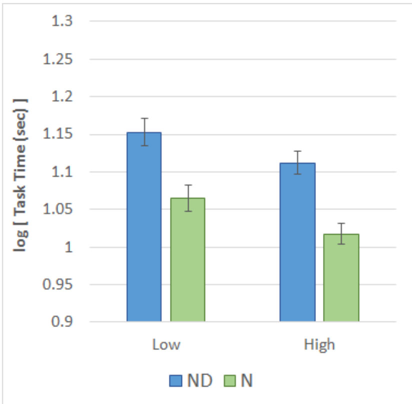
Figure 7.18 shows the relationship between between the effects of each factor on TT_{ND} compared to TT_N. All three figures show that task times were slower in *ND* conditions. Figures 7.18(b) and 7.18(c) show that task times for *Highlight Strength* and *Grid Size* followed the same general trend as the corresponding noticeability conditions. However, the difference between the TT_{ND} and the TT_N times was slightly larger in the higher-levels for each factor (i.e. for *High* strength, and for 64-item grids), suggesting that the effect of the distractors were greater in those cases.

Figure 7.18(a) shows that task times in the *ND* conditions did not follow the same trends as the noticeability (TT_N) and distraction (TT_D) task times. For example, the most noticeable technique (`Pulse`) had the second-lowest TT_{ND}, while the third-most noticeable (`Color`) had the lowest TT_{ND}. Similarly, the most distracting technique (`Pulse`) did not have the highest task time here (`Shooting Star` did instead). The only similarity between the distraction and *ND* data was that `Color` had the lowest task time in both. Pairwise comparisons confirmed that there was a significant difference between the TT_{ND} mean for `Shooting Star` and the TT_{ND} means for all other techniques.

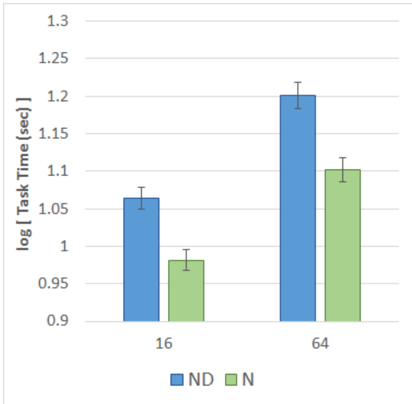
Figure 7.19 shows the interactions between the factors of the *ND* conditions. By cross-referencing these plots, several we can draw several interesting conclusions about the highlighting techniques.



(a) Highlight Type



(b) Highlight Strength

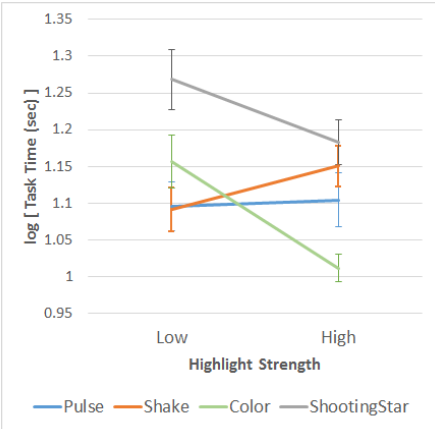


(c) Grid Size

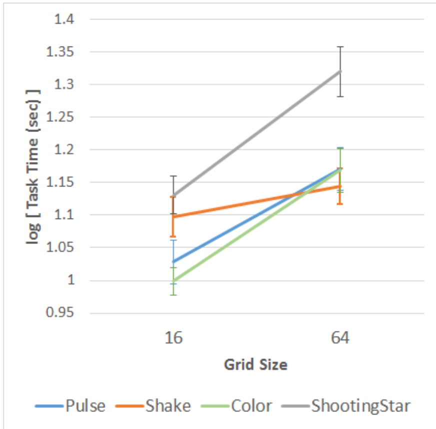
Figure 7.18: *Plots comparing the effects of the main factors for ND conditions (TT_{ND}) and noticeability conditions (TT_N). Log-transformed mean task times are shown here to make the differences clearer to see. Lower task times are better.*

In Figure 7.19(a) *Highlight Strength* did not appear to have any effect on `Pulse`. Figure 7.19(b) however shows that *Grid Size* did have an effect. These effects are explained by Figure 7.19(c): There was a significant crossover effect between `LPulse` and `HPulse` (i.e. the high and low instantiations of `Pulse`); while task times were slower for both in the *Grid Size* = 64 case, `LPulse` started lower but degraded by a larger amount, while `HPulse` started higher but degraded by a smaller amount. The "net" effect however in terms of *Highlight Type* × *Highlight Strength* is that there is no difference between the two instantiations of `Pulse`. **From these results, we can conclude that with small grids, `LPulse` is more effective (i.e. less distracting and more noticeable) in small grids, while `HPulse` is more effective in larger grids (i.e. less distracting).**

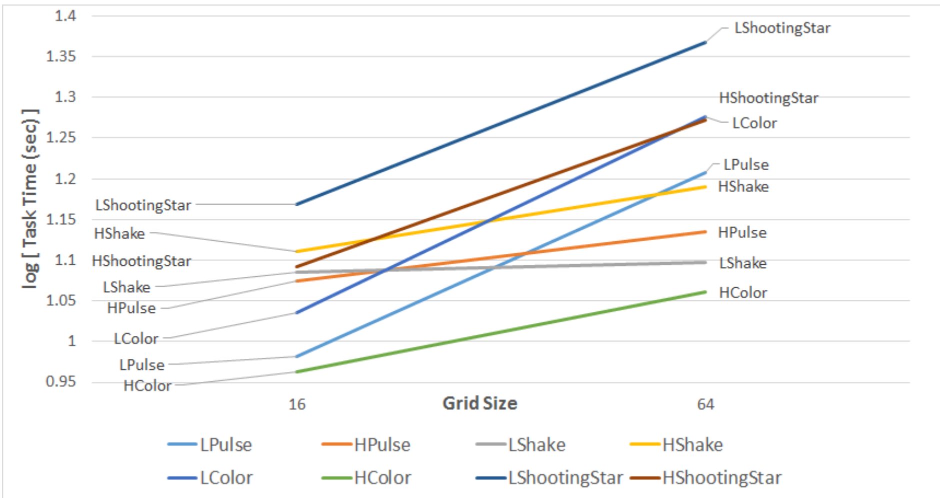
In Figure 7.19(a), there was a significant downwards trend for the *Highlight Type* × *Highlight Strength* interaction because task times across grid sizes for `HColor` were substantially lower than for `LColor` (as seen in Figure 7.19(c)). However, overall task times for `Color` increased with increasing *Grid Size* (as seen in Figure 7.19(b)). The performance degradation for `LColor` was more severe (resulting in `LColor` at 64-items having a high mean task time). However, since task times for `LColor` at 64-items in the noticeability (TT_N) cases also behaved similarly,



(a) Type × Strength



(b) Type × Grid Size



(c) (Type × Strength) × Grid Size

Figure 7.19: *Plots showing the interactions between factors for ND task time conditions. Log-transformed mean task times are shown here to make the differences clearer to see. Lower task times are better.*

the effect here was more likely to have been caused by `LColor` being harder to detect the effect of a distraction/incorrect decision more costly. **From this result, we can conclude that higher color contrast is needed for larger grid sizes**.

The other two techniques are relatively less interesting. In Figure 7.19, it can be seen that task times were `Shooting Star` were consistently high. This follows the same trends as for noticeability and distraction. Meanwhile, the differences between the different manipulations of `Shake` did not have any notable effects on the task times.

In conclusion, task time data for the *ND* conditions was useful for revealing some interactions between the highlighting techniques and their manipulations which may not have been apparent from the noticeability (TT_N) and distraction (TT_D) conditions.

7.3.2.5 Task Time Conclusions

These results show that *Task Time* is a sensitive measure for measuring the effects that highlighting techniques have on user performance. We have shown that for noticeability (TT_N), distraction (TT_D), and *ND* (TT_{ND}) trials, it was possible to find significant differences (in terms of Task Time) between the levels of the key factors being manipulated (e.g. *Highlight Type*, *Highlight Strength*, and *Grid Size*), with significant interactions between most combinations of these factors.

7.3.3 Movement Start Delay

Movement Start Delay (MSD) measures how long it took for the participant to start moving the mouse after the grid of candidate items appear. It is a secondary measure of user performance, as it is a subcomponent of *Task Time* (see Figure 7.3). In theory, this measure should represent the amount of time is required for the participant to start moving towards a highlighted item they have identified as being the target (or a potential target). This section addresses whether it is suitable for being used to analyse the noticeability and distraction of different highlighting techniques.

7.3.3.1 Relationship Between Movement Start Delay and Task Time

Figure 7.20 shows the relationships between the MSD times for noticeability, distraction, and *ND* times for all highlighting techniques. Several main conclusions can be drawn from this figure:

1. **The general $MSD_N \leq MSD_{ND} << MSD_D$ trend still holds.** That is, *MSD* can still be used to show that *ND* trials are still more similar to noticeability trials than distraction trials, and that user performance in distraction trials will still tend to be worse.
2. **Participants react faster when highlights are present**. MSD times for most of the distraction conditions are faster than for the *Control* condition, indicating that participants were responding to perceived targets faster than when there was no highlighting present.
3. **There does not appear to be any easily identifiable link or correlation between times for noticeability conditions and times for distraction conditions**.
4. **All mean MSD times are lower than those for Task Times**. As expected, the mean MSD times appear to be lower than their Task Time counterparts. This validates that there were no major data processing errors.

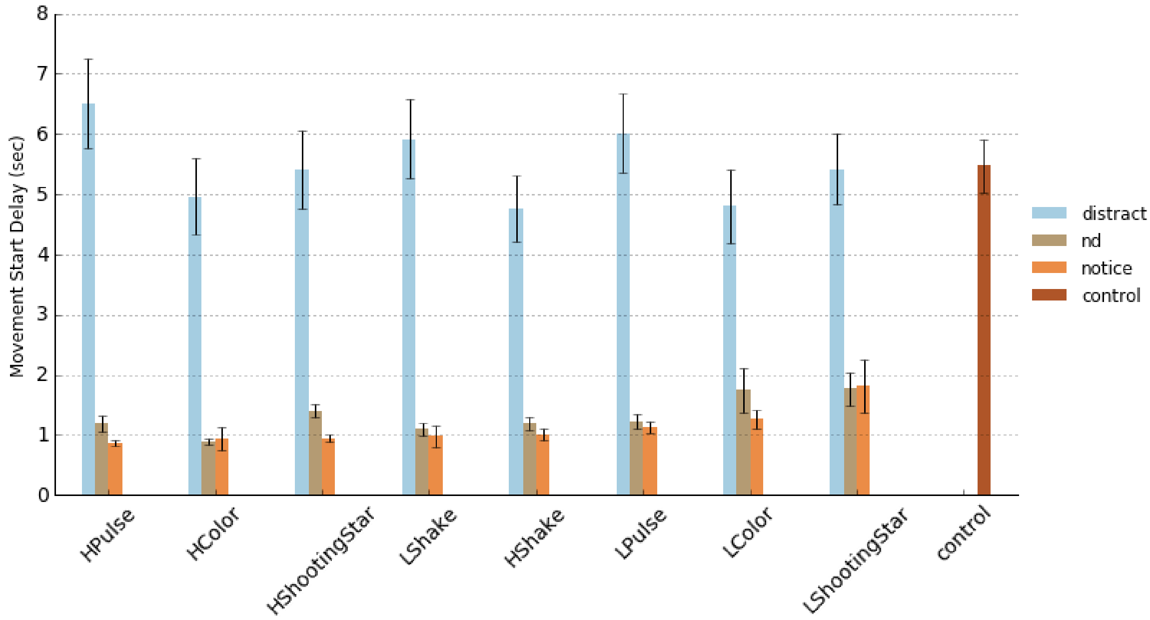
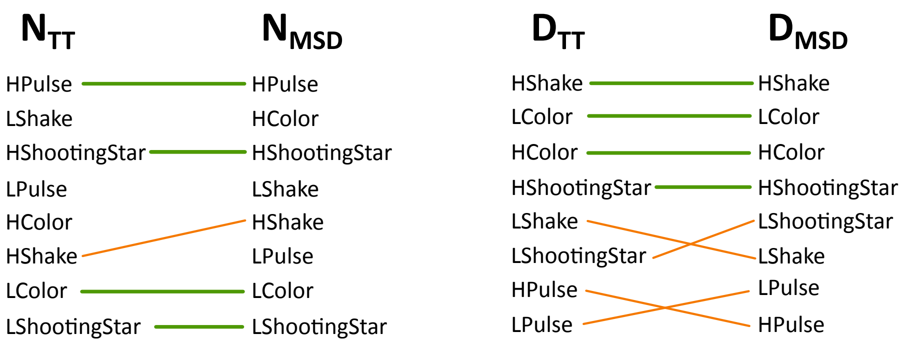


Figure 7.20: *Overview of how mean Movement Start Delay times vary between different combinations of highlighting techniques and stimuli conditions. The techniques are shown in order of increasing MSD noticeability times (lower times are better).*

From Figure 7.20, it can be clearly seen that the relative noticeability of different techniques (as measured using MSD) is different from the rankings derived from task time. This leads to the obvious question: how *did* MSD compare to task time? A comparison of the differences between these is shown in Figure 7.21.



(a) Comparison of how noticeable each highlighting technique is (from most to least)

(b) Comparison of how distracting each highlighting technique is (from least to most)

Figure 7.21: *Comparison of relative quality of highlighting techniques (from best to worst) in terms of noticeability and distraction, as measured from Task Time (X_{TT}) compared to Movement Start Delay (X_{MSD}).*

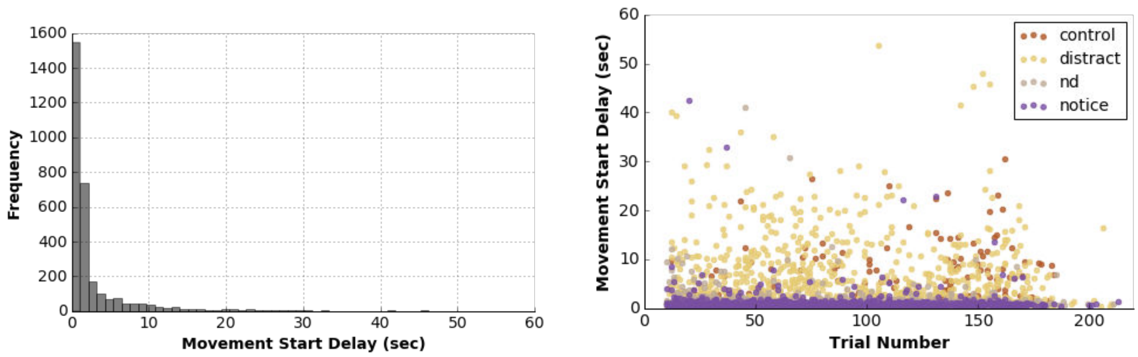
Figure 7.21(a) shows that apart from a few exceptions, noticeability rankings calculated from task time (N_{TT}) had little in common in noticeability rankings calculated from MSD (N_{MSD}). For instance, although `HPulse` was still the most noticeable technique, `HColor` was the second most noticeable (instead of `LShake`). Surprisingly, `HShootingStar`, `LShootingStar`, and

LColor (i.e. third, last, and penultimate respectively) had the same rankings in both the TT_N and MSD_N data.

Figure 7.21(b) in contrast shows that rankings for distraction were largely the same (or similar). For instance, the four least distracting techniques (HShake, LColor, HColor, HShootingStar) had exactly the same rankings with both measures. The four most distracting techniques however were very similar, with the order of two pairs of techniques swapped (i.e. LShake $\leftrightarrow$ LShootingStar, and HPulse $\leftrightarrow$ LPulse).

7.3.3.2 Distribution of Movement Start Delay Times

Figure 7.22(a) shows a histogram of the task times across all conditions. As can be seen, *MSD* is also a heavy-tailed distribution, with mean = 2.796 sec (min = 0.000 sec, max = 53.78 sec). For comparison, the corresponding parameters for *Task Time* were mean = 4.936 sec (min = 0.88 sec, max = 57.17 sec).



(a) Histogram of MSD frequencies (b) By condition type

Figure 7.22: *Plots showing overall distribution of Movement Start Delay times*

Of particular concern here was that the minimum *MSD* time was zero seconds. This was true in 2% of all trials (i.e. 71 / 3051). If all assumptions in Section 7.2 were true (i.e. that *MSD* accurately measures how long it takes for participants to notice/react to a highlight), that would imply that participants were able to instantaneously recognise the presence of highlighting techniques! However, it is well established [44, 94, 85] that there are delays of at least 0.1 seconds before humans are even aware of the presence of stimuli, let alone to start responding to it. Instead, this was more likely to have been the results of participants not following the procedure correctly by not keeping the mouse still.

Therefore, MSD does not appear to be a suitable metric for further analysis in our study. Given that MSD was a long-tailed distribution, it would have been necessary to again log-transform the data. However, the presence of zero-values and other low-values is problematic: although the problems with $log(0)$ being undefined can be solved by transforming the data using $log(1 + x)$ instead, the presence of the zero values casts doubt on the validity of the rest of the data – particularly for other low MSD values (e.g. the participant may have briefly paused their constant mouse movements when the stimuli appeared, so the first movement event may not in fact have been in response to the stimuli).

7.3.4 Eye Movements – Where was visual attention directed?

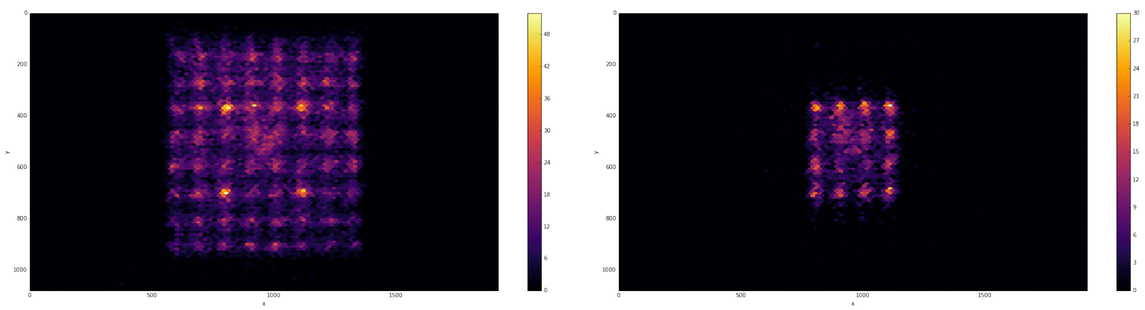
In addition to the performance measures reported above, we also collected and analysed eye tracking data. From the eye tracking data, we wanted to understand what effect highlights had on how participants performed their tasks. This lead to several key questions:

1. What effects did the experiment manipulations (e.g. the types of highlighting techniques used, and the configuration they were in) have on where visual attention was directed?
2. What was the relationship between the time taken for participants to first notice the target and how "noticeable" the target was (as measured using the N_T noticeability metric)?
3. What effect did distractors have on participant behaviour as they searched for the target? Specifically, what was the "main effect" of distractors? What made them distracting?

7.3.4.1 Where Was Visual Attention Directed?

Density heatmaps were generated to visualise the distribution of fixation points across the visual field. The colour of each bucket/cell in the heatmaps indicated the density of fixation points in that region. This was achieved by plotting the location of each fixation point across all trials and participants, then dividing the resulting 1920×1080 (i.e. Full HD resolution) domain into a grid of hexagonal bins (at a resolution of 100 bins horizontally, or 1 bin per $19.2 \times 19.2px^2$). Only fixations occurring after the *Onset Time* delay for each trial (see Figure 7.3) were included for this analysis.

Figures 7.23(a) and 7.23(b) show the distribution of fixations for 64 and 16 item grids respectively. In both figures, the grid layouts can be clearly seen from the patterns formed by the fixations (i.e. a large grid for 64-items, and a small grid for 16 items), with fixations clustered around the center of each item. Although some fixations did occur outside the grid profile, they were relatively infrequent; as a result, the rest of the plots presented here are cropped to only show the fixations that occurred within the gridded region.

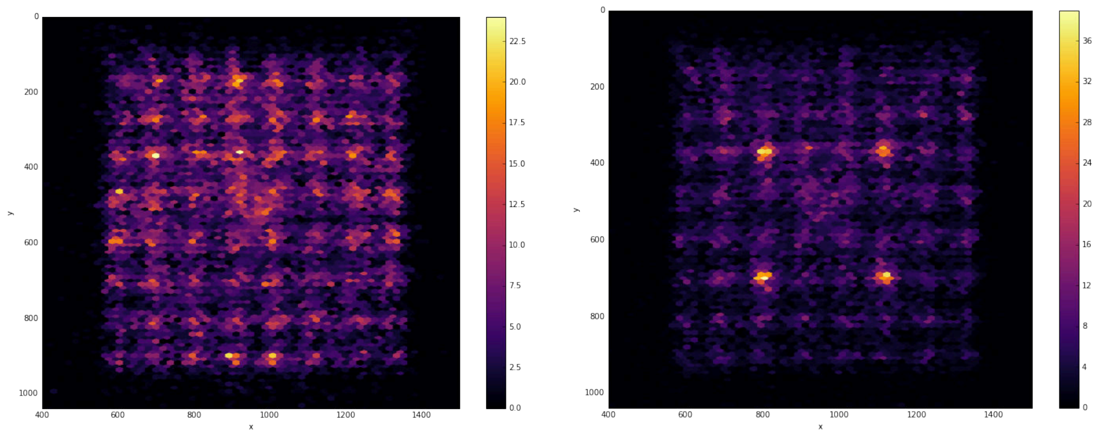


(a) N = 64 (b) N = 16

Figure 7.23: *Heatmap showing distribution of all the fixation points across all trials and participants, for grids with 64 and 16 candidate items*

Some items appear to have attracted more fixations than others – for instance, the four bright-coloured targets near the center of Figure 7.23(a) or the outer ring of items in Figure 7.23(b). While the pattern for the 16-item grid was to be expected (since all the targets and distractors appeared in that ring), the distribution for the 64-item grid was not as we had anticipated.

To investigate, the 64-item data was further partitioned using the *Distance* factor (see Figure 7.24). In theory, there should be significant differences between the distribution of fixations in the $Distance = 3$ and $Distance = 1$ cases – in the former, the targets should be in mid-far peripheral vision, while the targets should be closer to the focal point. This means that participants should have had less need to perform visual search across most of the field in the $Distance = 1$ case, as the targets were closer and should have been easier to find. Indeed, Figure 7.24(b) shows that the $Distance = 1$ case was the cause of the high fixation density on the Ring 1 items. In contrast, fixation densities in Figure 7.24(a) were more evenly distributed, with suggesting that participants spent more time scanning over all the items to find the target in those conditions. Due to the differences between these two conditions, they will be treated separately in the following analyses.



(a) N = 64, Distance = 3 (b) N = 64, Distance = 1

Figure 7.24: *Comparison of fixation distributions for 64-item grids when targets/distractors were in Ring 3 (Distance = 3) versus Ring 1 (Distance = 1). Note the difference in colour scales between these plots*

From these plots, several observations can be made about the nature of where visual attention was directed:

1. **The overall "brightness" of each plot is indicative of the amount of visual search effort that took place.** Brighter areas correspond to a higher density of fixation points, so the more areas that had higher densities of fixations can be interpreted as meaning that more areas had to be searched.

2. **Bright spots indicate areas where visual attention was directed most often**. There are two possibilities: 1) Visually salient items (i.e. highlighted items, or the targets) occurred at those points, or 2) Participants developed a bias towards examining those points. In Figure 7.24(b), it is likely that the four distinctive hotspots are caused by most of the targets appearing at those locations, given that there are not that many locations that can be used at that distance from the center of the field.

3. **The contrast between the bright spots and the overall brightness of the field are indicative of the ability of the highlighting techniques to achieve "pop out" effects**. For example, there is a pronounced contrast between the hotspots and other items in Figure 7.24(b), suggesting that the highlights/targets had a greater pop-out effect which reduced the need to perform costly visual search on the rest of grid (which had lower fixation densities as a result). In contrast, there is relatively low contrast between the hotspots and standard items in Figure 7.24(a), suggesting that participants often had to scan most of the items in the field to find the target (i.e. higher fixation counts in general), as the target item did not stand out as much. There are a few hotspots (e.g. the two bright spots between $x = [600, 800]$, and $y = [300, 500]$, and the two on the bottom edge) which have higher counts than the surrounding items – these appear to correspond with the locations of highlighted targets (see Section 7.7).

7.3.4.2 Effect of Highlighting Patterns on Visual Attention

As stated in Section 7.2.3, *Highlighting Pattern* refers to the experimental condition determining which items were highlighted (i.e. the target (*Notice*), a distractor item (*Distract*), the target and a distractor (*ND*), or none of the items (*Control*)). An obvious question to ask is what effect the different highlighting patterns had on where visual attention was directed.

Figure 7.25 shows a "*Fixations Matrix*" comparing the effects of the different highlighting patterns in each of the three different grid configurations (i.e. 64-Items with targets in the outer ring, 64-Items with targets in the inner ring, and 16-Items).

As with the Task Time results (Section 7.3.2), the *Notice* and *ND* conditions were quite similar to each other, while the *Distract* and *Control* conditions were similar to each other instead. Each of these pairs shared a similar pattern/distribution of fixations between them. For example, in the *Notice* and *ND* conditions, there was a "cross" shaped pattern in the "`N = 64, Distance = 3`" cases, and square-shaped hotspots in the "`N = 64, Distance = 1`" cases. Similarly, the *Distract* and *Control* conditions were quite similar to each other because in both cases, fixations are quite evenly distributed over the entire grid, suggesting that participants had to spend a lot of effort scanning through most of the items to find the target, thus explaining the higher task times for these conditions.

The main difference between the conditions in each pair was that the *Notice / Distract* cases were brighter overall (or were less contrasty) than the *Control / ND* conditions. For the *ND* conditions, the higher contrast between the "hotspots" and other items suggests that fixations were concentrated more on the highlighted items. This also suggests that the reason why the *ND* conditions were so quick to perform (and similar to the *Notice* conditions) was because effect of the highlights was still to reduce the size of the subset that needed to be considered.

The differences between the different highlighting patterns in the 16-item cases however were less pronounced, as the density patterns were quite similar for all four cases. The main difference between the 16-item cases was that the peak fixation densities (see the top-most value on the colour scales beside each plot in Figure 7.25) were different; these values were 6.4 for *Notice*, 15.0 for *Distract*, 5.0 for *Control*, and 10 for *N+D*.

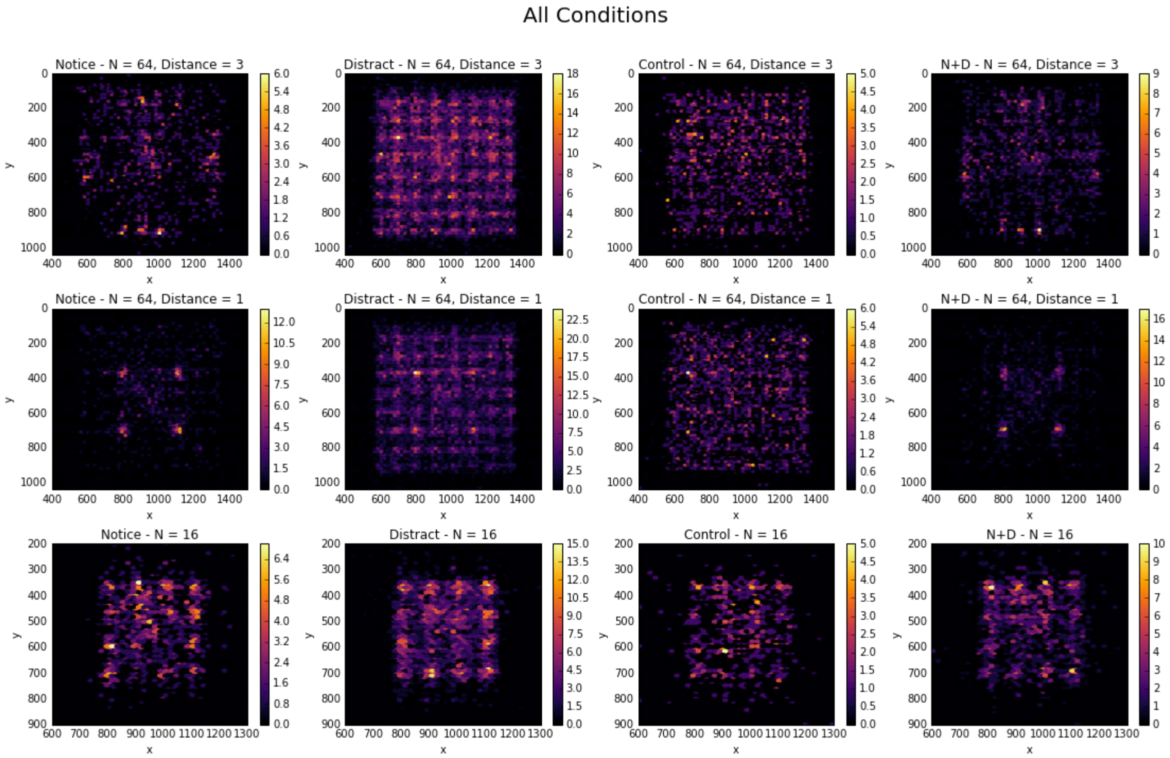


Figure 7.25: *Comparison of the effect of different highlighting patterns on where visual attention was directed. Note the different colour scales used between different plots.*

The peak fixation densities (PFD) tell an interesting story about where visual attention was directed in each case. For example,

- Across all grid size/distance configurations, PFD's were highest in the *Distract* conditions, suggesting that a possible effect of distractors was that they increased the frequency with which participants fixated on certain items.
- The second highest PFD's were for the *ND* conditions, which were almost double the PFD's for the *Notice* or *Control* conditions; this would make sense if participants had to look at the target and distractor items (both of which were highlighted, and differed in a subtle way) multiple times to verify that if they had identified the correct item.
- PFD in the *Notice* case for "`N = 64, Distance = 1`" were almost double the PFD's for the other two *Notice* configurations. Given that the PFD for the *ND* condition for this configuration was also higher than the others (i.e. 16 vs 10 for the others), and the fact that fixations in this configuration were mostly focussed on four distinct hotspots, the most probably explanation here is that the higher PFD's were because the target items occurred in fewer places for this grid configuration.

7.3.4.3 What did Participants Focus on First

The eye tracking analysis above focussed only on the *spatial* aspects the fixations (i.e. *where* did the fixations occur?) However, there is also a *temporal* component to the eye tracking data (e.g. *when* did the fixations occur, and *how long* did each fixation last?) In particular, we were interested in when the first fixations occurred, especially with regard to the target and distractor items. We assumed that each fixation point represents an area of interest that the participant was actively considering, rather than just being an "intermediate waypoint" that they just happened to rest their eyes on while saccading to their real target. Therefore, the first fixation that fell upon a highlighted item was important, as the time taken for the participant to notice the item could be considered a measure of how visually salient it was.

In the following few sections, we analyse three metrics of looked first (and when):

1. Time to First Fixation on *Any Item* (FFT_{Any})
2. Time to First Fixation on *Target* (FFT_{Target})
3. Time to First Fixation on *Distractor* ($FFT_{Distractor}$)

7.3.4.4 Time to First Fixation on Any Item

Figure 7.26 shows where the first fixation in each trial occurred after the stimuli appeared. As expected, most fixation points were still clustered around the center of the screen, where participants had been told to focus on. However, the distribution of fixations in each case was quite different. The "`N = 64, Distance = 3`" case least-resembled the final distribution (with no discernible target-based clusters), and had a small cluster around the center with a large spread outwards to the north-south-east-west directions. In contrast, in the "`N = 64, Distance = 1`" case, the majority of fixations were already focussed on the four hotspots which appeared prominently in Figure 7.23 for all highlighting patterns using this configuration; this suggests that the highlights had a significant pull on visual attention when placed Ring 1 in this configuration.

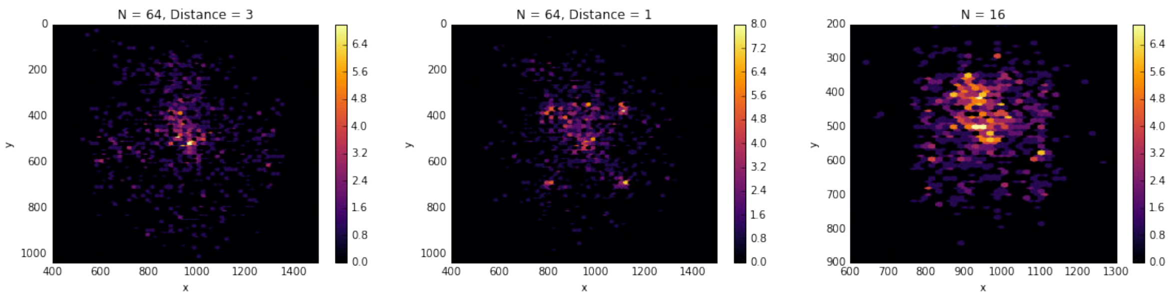


Figure 7.26: *Comparison of locations of the first fixation points in each trial. Note the different color scales used between different plots.*

Figure 7.27 compares the *time to first fixation on any item* (FFT_{Any}) data. This was the time taken for participants to focus on one of the points shown in Figure 7.26).

Analysis of the log-transformed FFT_{Any} data as a 3-factor within-subjects ANOVA (i.e. *Highlight Condition* × *Highlight Pattern* × *Grid Size*) found that there were only significant effects for the *Highlight Type* factor ($F_{7,133} = 2.418, p = 0.023$), and for the *Grid Size* factor ($F_{1,19} = 6.117, p = 0.023$). Post-hoc Tukey HSD tests showed a significant difference between the FFT_{Any}'s for `HShake` and `LShootingStar`, and between `HShake` and `Control`.

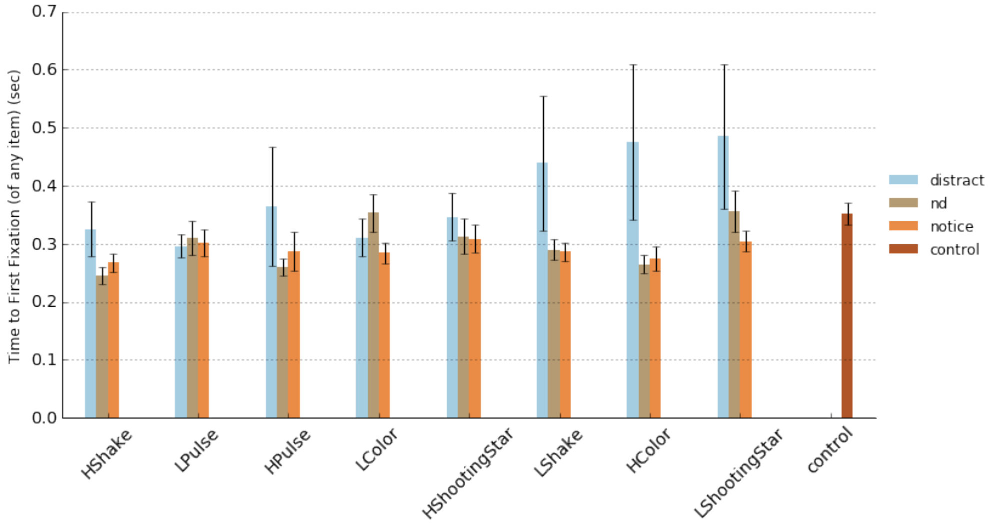


Figure 7.27: *Time to first fixation on any item. Sorted in order of increasing mean FFT_{Any} values for each Highlight Type (for all three highlighting patterns)*

These results suggest that the highlighting technique had an effect on how long it took for participants to start scanning for the target. A possible interpretation is that this indicates how noticeability/salience of each technique in peripheral vision. For example, the mean FFT_{Any} for `LShootingStar` was slower than in the `Control` condition (i.e. 0.3814s versus 0.3519s), as participants often had to wait for a short pause to see where the dot was travelling before being able to act. In contrast, `HShake` appears to have been quite detectable in peripheral vision, meaning that participants were able to quickly hypothesize where the target may be, resulting in a shorter delay before the first fixation occurred.

Interestingly, `HColor` was the second-worst technique, although participants rated it as being the most noticeable (see Section 7.3.1.2); this result is consistent with the notion that in our peripheral vision, we are less sensitive to differences in colour than the presence of motion. However, `LColor` was the fourth-best overall – ahead of `HShootingStar` and `LShake` – both of which were motion-based techniques. It is possible that the differences between the ordering of these techniques were just the result of noise, given that significant differences were not found between these any pair of these techniques.

7.3.4.5 Time to First Fixation on Target

FFT_{Target} is a measure of the *time taken for participants to look at the target item for the first time*. It is an important eye tracking metric, as it represents the first time in each trial that the participant noticed the target item. This is significant for several reasons:

1. In *noticeability* conditions, FFT_{Target} is the most direct measure of how long it took for visual attention to fall on the highlighted item, and is thus a measure of the visual salience of the highlighting effect. Highlighting techniques with lower FFT_{Target} times are therefore more visually salient, as visual attention focussed on the target earlier than in techniques where this time was higher.

2. In *distraction* conditions, FFT_{Target} is a measure of how much harder it was for participants to notice the target item as a result of the distractors. A larger value would indicate that the distractors had a greater negative effect on the participant's ability to notice the target.

Figure 7.28 shows the FFT_{Target} results. Overall, it took significantly longer to locate the target in the *Distract* and *Control* conditions, where the target was not highlighted. Compared to the *Task Time* values (Figure 7.11), there was a significant 1-4 second time difference between when participants first notice the target and when the target is successfully acquired.

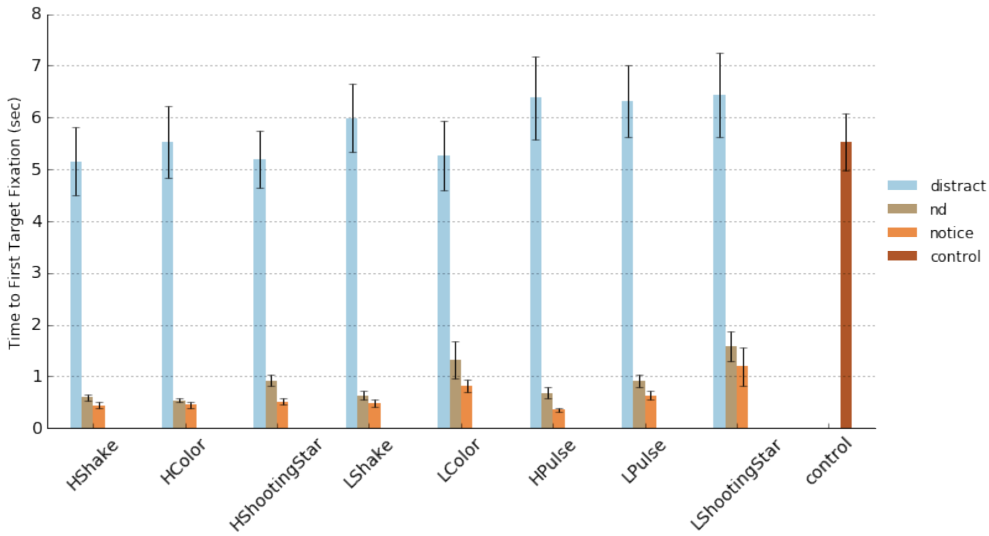


Figure 7.28: *Time to first fixation on the target item, for all highlighting conditions.*

Figure 7.29 focusses on just the conditions where the target was highlighted (i.e. *Notice* and *ND*). It shows that FFT_{Target} times for *ND* conditions did not follow the same trend as the *Notice* conditions: for example, in some cases (e.g. `HPulse`) the difference between the two conditions is larger than in other cases (e.g. `HColor`).

7.3.4.5.1 Time to First Fixations on Target – *Notice* Conditions

Analysis of the log-transformed FFT_{Target} times for the *Notice* conditions using 2-factor within-subjects ANOVA (*Highlight Condition* × *Grid Size*) found that, as expected, there were significant effects for both *Highlight Condition* ($F_{7,133} = 8.788, p = 0.000$), and *Grid Size* ($F_{1,19} = 4.819, p = 0.041$). However, there was no interaction between them ($F_{7,133} = 1.591, p = 0.143$).

Post-hoc Tukey HSD tests showed that there were significant differences between the means for the following pairs of highlighting techniques:

- `HColor` – `LColor`, `LShootingStar`

- `HPulse` – `LPulse`, `LColor`, `LShootingStar`

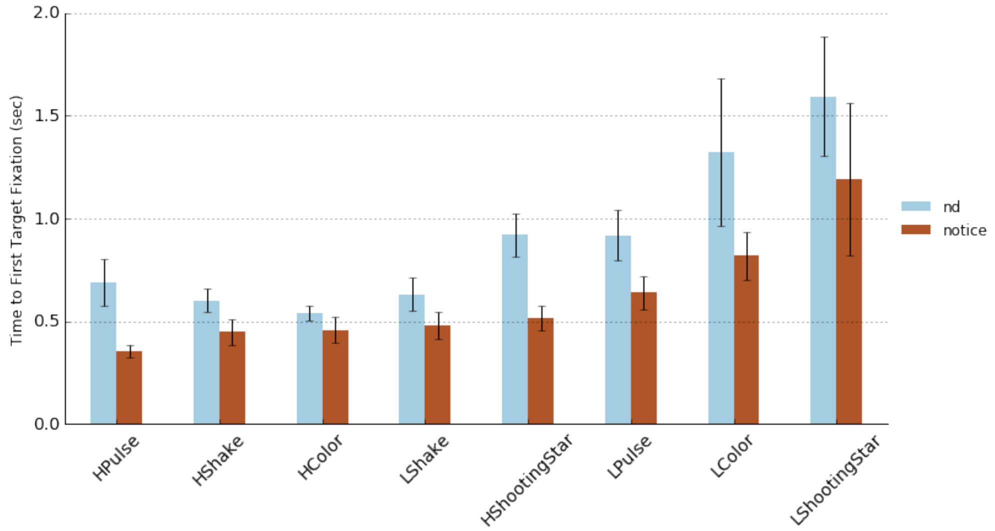


Figure 7.29: *Time to first fixation on the target item, for Notice and ND conditions only.*

- `HShake` – `LColor`, `LShootingStar`
- `HShootingStar` – `LShootingStar`
- `LShake` – `LShootingStar`

These results show that there was a significant difference between the high and low strength instances of each highlighting technique, *except* for High and Low strength `Shake`. Also, `LShootingStar` was significantly different from all high-strength techniques, as well as `LShake`. `LColor` was significantly different from all high-strength techniques *except* for `HShootingStar`.

7.3.4.5.2 Time to First Fixations on Target – *ND* Conditions

Analysis of the log-transformed FFT_{Target} times for the *ND* conditions using two-factor within-subjects ANOVA (*Highlight Type* × *Grid Size*) found that there were significant effects for both factors and also a significant interaction between them: *Highlight Type* ($F_{7,133} = 8.250, p = 0.000$), *Grid Size* ($F_{1,19} = 9.306, p = 0.007$), and *Highlight Type* × *Grid Size* ($F_{7,133} = 2.585, p = 0.016$).

Post-hoc Tukey HSD tests found significant differences between `LShootingStar` and every other highlighting technique – i.e. `LShootingStar` – `HColor`, `LShootingStar` – `HShake`, . . . , `LShootingStar` – `HShootingStar`. This indicates again that the `LShootingStar` technique was very ineffective.

Figure 7.30 compares the *ND* conditions for *Highlight Type* × *Grid Size*. It shows that while there was little difference between grid sizes for some techniques (i.e. `HColor`, `LShake`, `HShake`, and `HPulse`), others showed significant differences (i.e. `HShootingStar`, `LPulse`, `LColor`, and `LShootingStar`). High FFT_{Target} times indicate that the distractors had a greater negative effect on task performance. Thus, given that the techniques with the largest differences are some of the least noticeable (i.e. `LColor` and `LShootingStar`), these results suggest that less noticeable techniques were more disruptive/distracting in *ND* conditions, as looking at the wrong highlighted item drew participants further away from the target, further increasing the difficulty of finding it. In contrast, those with little difference between grid sizes were less affected as they were sufficiently noticeable that incorrect decisions could be quickly rectified.

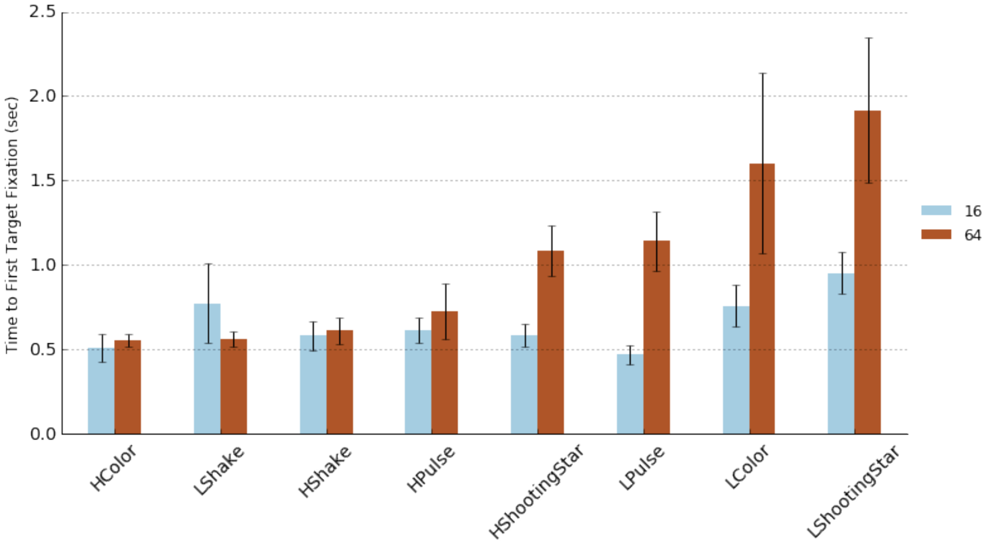


Figure 7.30: *Time to first fixation on the target item, showing the interactions between Highlight Condition and Grid Size for ND conditions. Highlighting types have been sorted in increasing order based on the 64-item ND data.*

7.3.4.5.3 Time to First Fixations on Target – *Control* and *Distract* Conditions

Figure 7.28 shows that in the *Control* and *Distract* conditions, FFT_{Target} was significantly higher than in the *Notice* and *ND* conditions. The only difference between the former and latter pair was the target item was not highlighted in the *Control* and *Distract* conditions. This lead to us to wonder whether there was any difference between the FFT_{Target} times for the *Control* and *Distract* conditions due to the different highlighting techniques being used as distractors.

Figure 7.31 compares the FFT_{Target} times for *Distract* and *Control* conditions. Analysis using 2-factor within-subjects ANOVA (*Highlight Condition* × *Grid Size*) found that there were no significant differences between the different types of highlighting techniques ($F_{7,133} = 1.525, p = 0.164$). The only differences in these conditions existed due to the difference in *Grid Size* at a significance level of $p < .001$ ($F_{1,19} = 89.267, p = 0.000$).

These results indicate that the different types of distractor did not affect how long it took participants to first notice the target item. This means that any differences in *Task Time* would have occurred after the target was first spotted – for instance, if the presence of the distractor made it harder to drag the cursor towards the target despite having seen it, or if the distractor causes the participant to doubt their judgement.

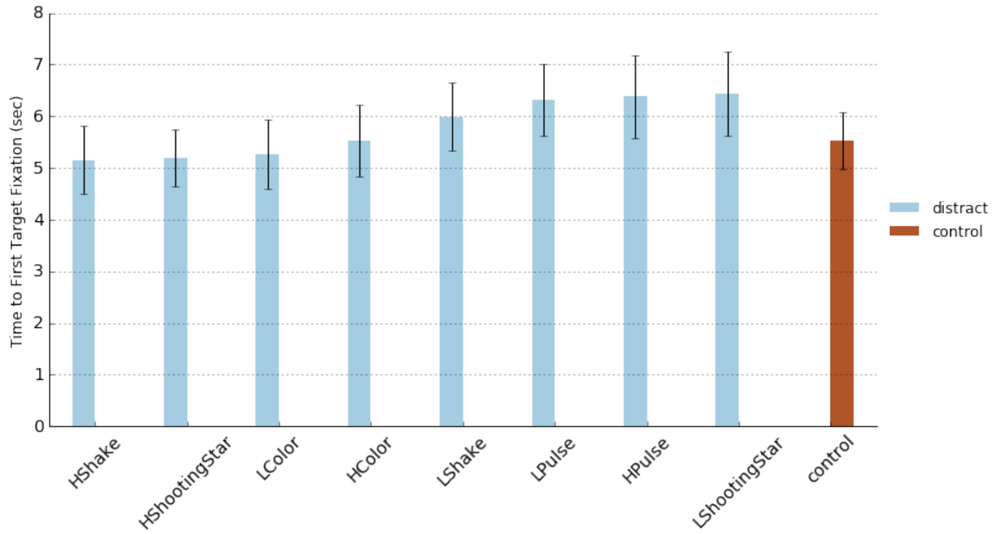


Figure 7.31: *Time to first fixation on the target item, for Distract and Control conditions only.*

7.3.4.6 Time to First Fixation on Distractor

The *time to the first fixation on a distractor* ($FFT_{Distractor}$) measures how long it took for participants to first look at the distractor. We expected that the findings of this metric were likely to fall into one of the following two scenarios:

1. Assuming that participants only needed to sense the presence of a highlighted item (in peripheral vision) to have their attention quickly drawn towards it, then in theory, this metric should be very similar to the FFT_{Target} metric examined in the previous sections.

2. However, if participants learned to be wary of or even ignore the highlights, then it becomes harder to predict or explain what participant behaviour may be like in terms of $FFT_{Distractor}$.

   This scenario may occur because the presence of a highlight was only sometimes associated with successful detection of the target. Thus, the Payoff Matrices (participants were implicitly using when responding to the highlights) would have been weighted to give less importance to immediately investigating highlights, as there is an equal (if not greater) perceived cost to attending to a highlight only to find that it is a distractor.

7.3.4.6.1 Time to First Fixations on Distractor – *Distract* Conditions

Figure 7.32 compares the effects that different highlighting techniques and grid sizes had on $FFT_{Distractor}$. The graph shows that there appeared to be an interaction between *Highlight Condition* $\times$ *Grid Size*. For example, $FFT_{Distractor}$ was lowest in the 16-Item `HPulse` condition, but was the highest in 64-Item condition for the same highlighting technique. Another interesting example was the `HColor` technique, which had the highest $FFT_{Distractor}$ time in the 16-Item case, but had the fourth-slowest $FFT_{Distractor}$ time in the 64-Item case. The technique with the smallest difference between the 16 and 64 item grids was `HShake`.

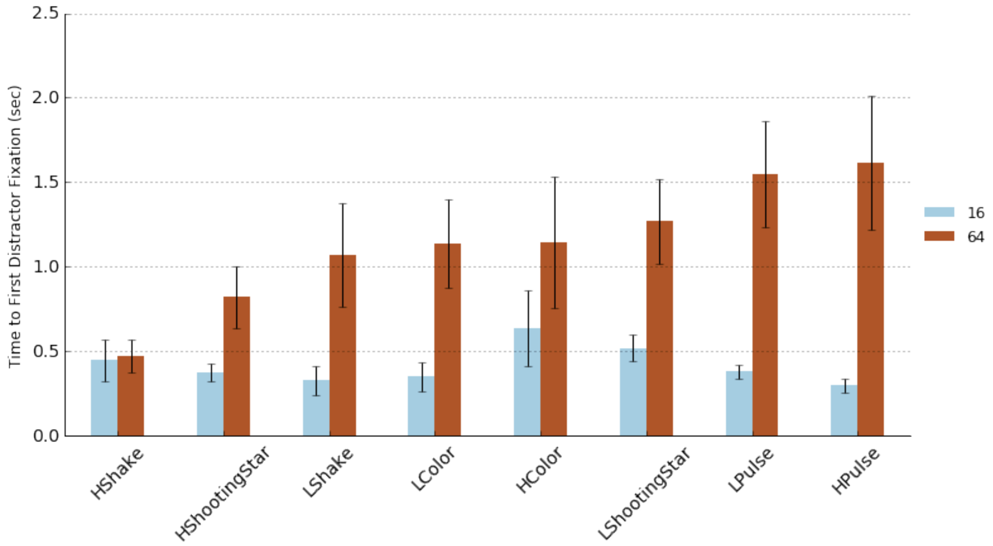


Figure 7.32: *Time to first fixation on the distractor item, showing the interaction between highlighting types and grid size in Distract conditions.*

Analysis of the log-transformed $FFT_{Distractor}$ data using 2-factor within-subjects ANOVA (*Highlight Condition* $\times$ *Grid Size*) confirmed that there were significant effects for both *Highlight Condition* ($F_{7,133} = 2.647, p = 0.014$) and *Grid Size* ($F_{1,19} = 29.915, p = 0.000$). There was also a significant interaction between these factors ($F_{7,133} = 2.712, p = 0.012$). Post-hoc Tukey HSD tests found significant differences between `HShake` – `LPulse`, and `HShake` – `LShootingStar`.

7.3.4.6.2 First Fixations on Distractor – *ND* Conditions

Figure 7.33 shows the effect of *Highlight Condition* $\times$ *Grid Size* on $FFT_{Distractor}$ for *ND* conditions. Analysis using 2-factor within-subjects ANOVA of this data found *no significant effects or interactions* between these factors: $F_{7,133} = 0.956, p = 0.466$ for *Highlight Condition*; $F_{1,19} = 1.074, p = 0.313$ for *Grid Size*, and $F_{7,133} = 1.429, p = 0.199$ for the interactions between these.

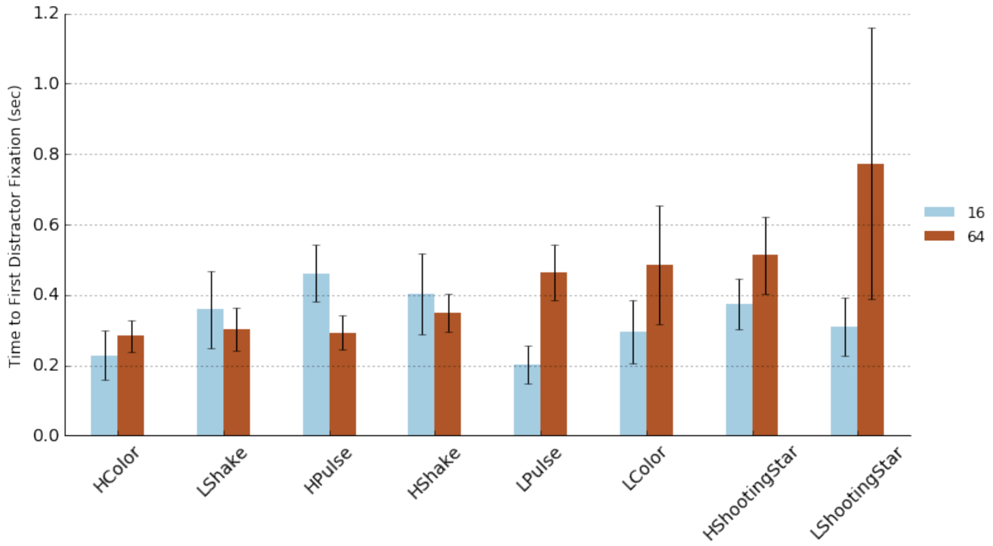


Figure 7.33: *Time to first fixation on the distractor item, for ND conditions.*

Notably, the relative order of the FFT_{Target} and $FFT_{Distractor}$ times in the 64-Item *ND* times were very similar. Figure 7.34 shows that the relative order was exactly the same for the first two places (`HColor` and `LShake`) and the last place (`LShootingStar`); it also shows that for two pairs of places (i.e. the `HShake` – `HPulse` pair, and the `LPulse` – `LColor` pair) the relative order was similar.

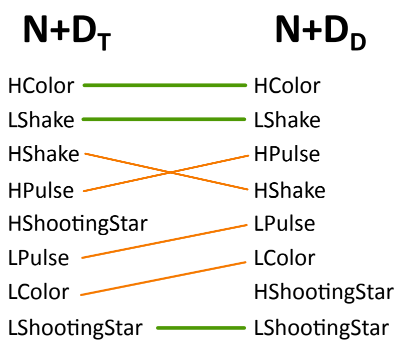


Figure 7.34: *Comparison of the relative order of First Fixation times on Target and Distractor items in 64-Item ND conditions.*

Another notable observation is that `HColor` was noticed the fastest for both the target and distractors in *ND* conditions. This is notable, as it is one of the few times where a performance-based metric identified `HColor` as being the most noticeable (as rated by the subjective experience metrics).

7.3.4.6.3 First Fixations on Distractor – *Distract* versus *ND* conditions

Figure 7.35 compares the $FFT_{Distractor}$ times for *Distract* and *ND* conditions. It shows that the $FFT_{Distractor}$ times in the *ND* conditions were generally lower than in the *Distract* conditions.

However, there was no clear relationship between these times.

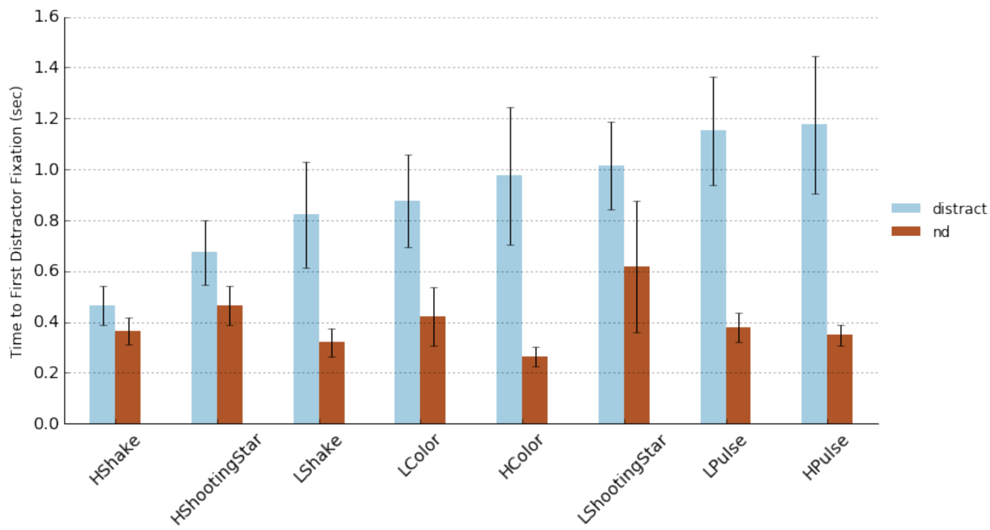


Figure 7.35: *Time to first fixation on the distractor item, for Distract and ND conditions.*

7.3.4.7 Relationship Between First Target Fixations and Task Times

The First Fixations on Target metric (FFT_{Target}) represents the first time that participants became aware of the location of the target. This raises the question: How long did it take for participants to complete the task once they had first encountered the target? From the other first fixation metrics, it was found that participants looked at target items earlier than distractor items. This suggests that instead of initially out-competing the target, distractors may have worked by making it harder to complete the task once the target was found.

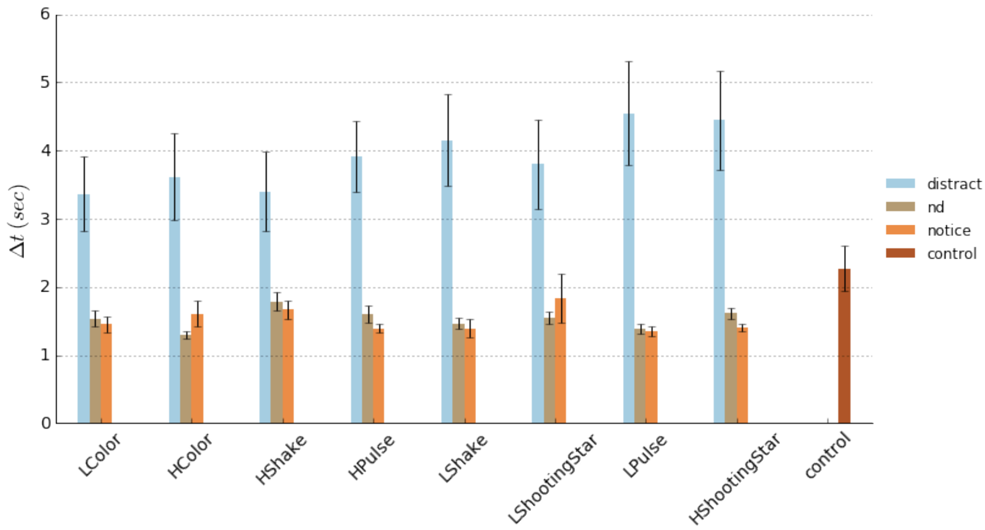


Figure 7.36: *Relationship between Task Time and FFT_{Target}*

Figure 7.36 shows the "Δt" values for each *Highlight Condition* × *Highlight Pattern* combination. The Δt metric used here is defined as:

$$\Delta t = TaskTime - FFT_{Target} \tag{7.5}$$

Figure 7.36 shows that, in general, the Δt values followed the same general pattern as the other metrics: the Δt values for the *Notice* and *ND* conditions were the lowest and most similar to each other, while the *Distract* conditions had the highest values. The *Control* condition was again higher than in the *Notice* and *ND* conditions, but that may have been due to the increased difficulty of the task (e.g. some participants may have momentarily lost track of the target, despite having spotted it already). However, there was a notable difference from the other metrics: in the *Control* condition, Δt was significantly lower than the *Distract* conditions by 1-2 seconds. This is further evidence that the main effect of the distractors was that they made it harder to complete the task.

7.3.4.8 Summary of Eye Tracking Results

The results show that there were significant differences between the fixation patterns when different highlighting techniques were used in different ways (i.e. to help the user, distract the user, or both).

The density heatmaps (one per grid configuration × highlight pattern) were a useful tool for visualising and exploring the distribution of fixation points (and thus the overall patterns of where participants were focussing their visual attention). It was surprising to note how different the distributions were (in particular for the *64-Item, Distance = 1* conditions).

In general, the eye tracking supported the findings from the *Task Time* data. It was also more useful than the *Movement Start Delay* measure could have been, as it is less likely that differences between how participants approach the visual search task (e.g. whether they continuously moved the mouse while searching in a bid to keep track of what they were considering) could end up biasing/masking trends in the data.

The most useful and interesting metrics were found to be:

- Time to First Fixation on Target (FFT_{Target})
- Time to First Fixation on Distractor ($FFT_{Distractor}$)
- Time between First Fixation on an Item and Task Times (Δt)
- Peak Fixation Densities (for the Heat Maps)

7.4 Discussion

7.4.1 Summary of Findings

The experiment results presented show that our experiment method was effective at measuring performance differences between different highlighting techniques and of factors which can influence the noticeability and distraction characteristics of those techniques (**H1.1**).

Overall, we found that there were many complex interactions between the highlighting techniques and how participants responded to them, as the relative ordering each technique (from best to worst) varied from metric to metric. This was particularly pronounced with the eye tracking data (in the Time to First Fixation metrics).

There were however several findings were consistent across the metrics:

- `HPulse` was usually the "best" technique (or within the top few techniques)
- `LShootingStar` was consistently the "worst" technique across all measures
- `LColor` was often the second worst technique
- `HShake` and `HColor` were quite effective overall. However, in general, `HColor` performed worse than the subjective ratings provided by participants suggested (it was rated the "best/most effective" technique).

Of the performance metrics used, most were found to be sensitive measures of human performance with highlighting techniques. The notable exception here was *Movement Start Delay*, which was less successful than expected as participants did not always properly follow the expected procedure.

7.4.1.1 Outcomes for Main Hypotheses

This study investigated the following main hypotheses, with the following results:

- **H1.2 – Measures of noticeability do not increase monotonically with increasing distraction** – *(Result: Confirmed ✓)*
- **H7.1 – Rankings derived from noticeability and distraction analysis directly correspond with subjective experience responses** – *(Result: Rejected ×)*
- **H7.2 –** $\boldsymbol{TT_N \leqslant TT_{ND} \leqslant TT_D}$ – *(Result: Confirmed ✓)*
- **H7.3 – In distraction conditions (i.e. only the distractor item is highlighted), participants will look at the distractor before looking at the target.** – *(Result: Confirmed ✓)*
- **When there is only a single item highlighted (i.e. the target in Notice conditions, or the distractor in Distract conditions), there should be no significant difference between the first fixation times.** – *(Result: Rejected ×)*

These results are notable, as apart from **H7.2** and **H7.3**, all the other results were contrary to commonly held beliefs about the nature of noticeability and distraction.

7.4.2 What do we learn about highlighting?

We found that the relative effectiveness of different highlighting techniques varied from metric-to-metric. The only exception here was `LShootingStar`, which was consistently ranked as being the worst/least effective technique. For all other techniques however, their relative ranking would fluctuate (sometimes substantially). This suggests that there many complex interactions exist between the different highlighting effects and the way that participants respond to these. From this initial study, there is not enough data yet to begin to understand what these factors may be. Further follow-up studies are required to investigate these issues further.

Overall, the most sensitive metrics/measures of human performance were: *Task Time*, *Time to First Fixation on Target* (FFT_{Target}), *Time to First Fixation on Distractor* ($FFT_{Distractor}$), and the *Time difference between* FFT_{Target} *and Task Time* (Δt). Less useful though were the *Time to First Fixation on Any Item* (FFT_{Any}) and *Movement Start Delay* (MSD) metrics. These latter two metrics were both concerned with events that occurred near the start of the trials, but turned out to be less useful than expected for several reasons: 1) Participants did not always follow the instructions (e.g. after engaging the left-mouse button, they would constantly move the mouse, or would move the cursor to the edges of the screen), making "first event" analyses not very useful, and 2) In many cases, the first movement/fixation was not directed towards either the target or the distractor.

We also found that the *Highlight Strength* manipulations sometimes had unexpected effects on noticeability and distraction. For example, `LShake` was found to be more noticeable than `HShake`: usually the lower-strength instantiation is less noticeable, but here it ended up being more noticeable. In the case of `Shake`, the higher speed movements in `HShake` could have made it harder to clearly see the corners of the item, thus making it harder to distinguish whether the highlighted item was the target. Another surprising example was with `LPulse` and `HPulse`: `LPulse` was less noticeable but more distracting than `HPulse`.

7.4.3 Novelty and Scope of Contributions

In this study, we have developed a novel methodology for measuring the noticeability and distraction characteristics of different highlighting techniques that to our knowledge has not been attempted in the HCI literature before. Similar experiment setups have been attempted in the psychology literature, but those attempts had different aims and used different metrics/measurement techniques (e.g. tracking three dimensional finger movement trajectories versus tracking mouse and eye movements).

In particular the key differences between the study presented and prior work in the HCI and psychology literature are:

- Our method attempts to empirically measure both noticeability and distraction effects in the same experiment, and using similar metrics.
- Our study compared how performance-based and subjective experience metrics ranked highlighting techniques in terms of noticeability and distraction. This was not done in prior studies as there were no performance-based metrics that could be used.

- Our study used eye tracking data as one of the metrics used to characterise human behaviour. We also compared the effects of the two sets of metrics.
- Our study included a range of different common highlighting techniques, and evaluated each of these techniques at more than one level of intensity. Instead of being completely arbitrary choices, these instantiations were chosen by taking a pair of extreme values (one low and one high) from each of the key parameters of each highlighting technique

7.4.3.1 How does this compare to prior techniques used in HCI?

There have been several prior studies of highlighting techniques in the HCI literature (see Section 3.3). Notable examples include the studies of Bartram et al. [29], Davies and Beeharee [57], and Kieras and Hornof [99].

Many of the prior studies in the HCI literature used a dual-task setup, where participants had to play some form of game (primary task) while responding in some way to highlights/notifications (secondary task, for characterising noticeability). Distraction however was rarely investigated or measured. In the studies where the distraction effects of highlighting techniques were studied (e.g. Bartram et al. [29]), distraction was only quantified as a subjective measure.

Bartram [28] justified the use of these dual-task setups (along with using only subjective measures of distraction) by arguing that they may be more "ecologically valid", as the highlighting techniques would be encountered within the environments where they would normally appear, and with more realistic workloads (e.g. there is a task that participants are actively focussed on accomplishing) than could be accomplished using a "simpler" task. We argue however that without a more solid understanding of the basic perceptual factors (independent of other tasks), it is premature to begin evaluating the techniques in a variety of complex task environments. First, we do not necessarily understand how well the subjective measures of distraction correspond to user performance/behaviour, so simply assuming that the subjective measures realistically/reliably capture the full story is a big assumption. (As discussed later in Section 7.4.4.4, the wording of the subjective experience questions can have potentially severe effects on the results). Second, it becomes harder to generalise or build predictive models about how the techniques may perform in a new environment/use case, due to methodological differences between studies. Third, it becomes harder to isolate the effect(s) of the highlighting techniques from any effects that were caused instead by the primary task being performed.

One disadvantage of our method for measuring noticeability over the methods used in the prior HCI literature is that we are only measuring noticeability for primary-task performance (i.e. the user is already searching for the target, so noticeability here refers more to whether the highlights assist the primary task). In contrast, the dual-task approach measures how well the highlights can redirect the user's attention away from their primary task towards a secondary source of information that is competing for the user's attention (e.g. a notification or alert generated by the system). In this latter case, even though the user's attention is being redirected away towards another task/focus, the information they gain may indirectly be useful for improving their primary task performance at some point in the near future. Both cases are equally valid types of noticeability; they simply represent two different use cases for highlighting techniques: 1) *As a passive visual hint or naviation aid* (this study), and 2) *As a notification or alert* (traditional dual-task studies). Therefore, it may still be useful to use dual-task paradigms for studying noticeability effects in isolation, and to combine those results with the results obtained using our method when evaluating highlighting techniques.

7.4.4 Limitations and Sources of Error

7.4.4.1 Measurement Errors

Despite our best efforts to minimise or eliminate measurement errors, it was inevitable that some of these still existed in the final experiment.

First, there was the problem of fatigue. It was necessary to have over 150 trials to accommodate all the conditions we needed/wanted to include. The tasks were also configured to make it hard to identify the target item. As a result, participants had to focus intensely in a significant proportion of the trials. The visually taxing nature of the experiment meant that many participants experienced fatigue during the experiment.

Second, there was the problem of learning effects. Despite the measures taken to minimise these (see next section), it was impossible to completely eliminate the potential for this to affect the experiment results in some way. Also, there was the remote possibility that despite our best efforts, participants did subconsciously learn to expect certain highlights/targets to appear in certain locations.

Third, at the start of each trial, the cursor was automatically recentered to ensure that all paths would start from a consistent location. This was done to minimise the chance that differences in initial starting position for the cursor would affect the results. However, despite this measure, some participants would constantly move the mouse, or would deliberately swipe the cursor to an area of the screen where they were anticipating to be "unpopulated" (i.e. often the lower left or right corners). This is despite these participants having already been instructed to not move the cursor until they had seen the target.

7.4.4.2 Measures taken against learning effects

The method had to be modified several times to mitigate learning effects. Specifically, we had to reconsider where the targets and distractors were appearing, while ensuring sufficient separation between the target and distractor. For example, in earlier versions of this experiment, participants were able to anticipate where the target might be located due to the limited number of target positions.

Several other measures were taken to minimise learning effects, including changing the ordering of the conditions, target locations used, and the length of the delay between trials. Conditions were randomly ordered instead of performing blocks of trials for a single technique in order to reduce possible fatigue.

The order of trials for the different levels of Target Location, Stimuli Condition, and Field Density factors were also randomised. This was done to reduce anticipation of the next stimulus configuration. However, it was possible that despite these measures, participants still developed intuitions about target/distractor locations. We found that some of the participants would start each trial by moving the cursor to the midpoints of the lower two quadrants; given that those same participants developed a habit of moving the cursor to an "off the grid" position whenever the `Shooting Star` condition appeared (thus minimising the distractions caused by the dots traversing over a large part of the grid), it seems reasonable to believe that these behaviours were linked, and that some kind of learning effect (e.g. to minimise the disruption caused by the distracting highlighting techniques) were occurring.

At the start of each trial, there was a random delay before the stimuli appeared. This was done to prevent participants from learning to anticipate when the stimuli would appear. Without this delay, participants would sometimes start moving in preparation for hitting the target "when it appeared", thus invalidating the *Movement Start Delay* measure. In retrospect, these measures may not have been effective enough – longer delays may have been needed to prevent participants from trying to move.

7.4.4.3 Accuracy of the Eye Tracking Data

In general, the accuracy of eye tracking data depends a lot on the accuracy of the initial calibration, as well as whether the participant's posture changed significantly enough that the calibration was invalidated. For the majority of participants, calibration accuracy did not appear to be an issue (as verified pre-experiment, and by visual inspection of the fixation data per participant).

There were however a few participants where tracking accuracy problems arose. For one participant, despite multiple attempts at obtaining an accurate calibration, we were unable to do so. For two participants (including the aforementioned one), there were also unexplained tracking failures, where for multiple consecutive trials, the eye tracker could not locate the participant's eyes.

7.4.4.4 Weaknesses of the Subjective Experience Measures

A key weakness of how the subjective experience measures were collected is that there may have been some ambiguity in the terminology used. For instance, we asked participants to rank the highlighting techniques in terms of how "noticeable" they were. However, it is possible that some participants may have had different interpretations about whether they were supposed to be rating the noticeability of the *target* (as a result of the highlighting technique being applied to it), or whether we were interested in the *highlighting effect* itself. The difference between these two interpretations is that the former considers issues of suitability for purpose/usability, whereas the latter only considers the "raw" perceptual response (i.e. "how strongly did that thing catch my eye").

In most cases, there is likely little difference between these two interpretations of noticeability. However, if there was an interaction occurring between the highlighting technique and the target cueing mechanism (e.g. the motion of the highlighting technique makes the true shape of the target hard to discern), it is possible that the technique may have been rated higher for "raw" noticeability (i.e. the motion itself is really salient), but would have been rated lower for utility/suitability instead (i.e. because the highlighting technique's motion made it hard to see if the highlighted item was the target, it was harder to tell whether the item was the target or a distractor, therefore making it less noticeable than other techniques).

It is also possible that "distraction" may have been interpreted slightly differently by different participants. Bartram [28] avoided using the term "distraction" directly in their questions, to ensure that participants were using a specific definition (based on the "attentional pull" of the technique). Instead, they had one question addressing the attentional pull effect, and another on how irritating/annoying the effect was [28]. The single question used here may have conflated the two.

The possibility of different interpretations first came to our attention when one of the later participants asked for clarification about which interpretation we wanted them to use when answering the questions. By that stage, it was already too late to change the question wording.

It may also be argued that it would have been useful to have participants rank the noticeability and distraction of each technique in absolute terms (e.g. as "strength" of N or D on a Likert scale). That way, there are subjective experience measures which could be compared against different studies. We did not do this due to considerations about the overall length of the experiment (i.e. a minimum of 163 somewhat challenging visual search tasks, taking around 30-35 minutes in total). Instead, we intended that our rank-based measures would force participants to discriminate between techniques, potentially aiding experimental sensitivity.

7.4.4.5 Sources of the differences between Subjective Experience Measures and Task Performance Metrics

Figures 7.9(a) and 7.9(b) show the relationship between noticeability and distraction, as measured from the *Task Time* (i.e. the first figure) and from the subjective rankings reported by participants (i.e. the second figure). However, there are some clear differences between the findings of the two datasets.

There are several possible explanations for this discrepancy. Given the issues we had with the ambiguity of the text, there is a possibility that participants were misinterpreting the questions. For example, there may have been different interpretations for whether we were asking about how easily the highlighting effect could be noticed, or whether we were asking about how easy it was to notice which item was the target as a result of the highlighting. Similar problems may have existed for distraction too, though it is not immediately clear what alternative interpretations participants may have been using. One of the most perplexing observations was that some participants rated the Control condition as being "more distracting" than some of the highlighting conditions.

It is also possible that some participants may have been responding to the situations given in the example images (with just two example items in close proximity), as opposed to responding based on their in-experiment experiences. Although the images were added as a cue for participants for identifying the techniques (since we never specifically assigned any participant-visible names for these), their very presence may have lead to participants making their rankings based solely on what was displayed in the images. This would have particularly been an issue if one technique works really well in a small grid, whereas its comparative performance in a much larger grid is significantly different.

Another possibility is that the objective measures of N and D may include confounding factors. Specifically, the "Noticeability" measure (N_T) is derived from the *(Total) Task Time* metric:

$$N_T = TaskTime = T_{PO} + T_{ID} + MT \tag{7.6}$$

where T_{PO} is the time taken for participants to actually notice the highlighting effect, T_{ID} is time time taken for participants to determine whether the highlighted item is the target (i.e. are the corners actually rounded), and MT is the time needed to point to the target. Ideally, N_T would be a close approximation of T_{PO} (i.e. $T_{PO} + c$, where c is a near-constant and negligible factor). That is, the Noticeability metric effectively only measures the time needed to detect the pop-out effects of the highlight (as intended). However, in practice, on fast moving targets (e.g. with the `HShake` and `LShake` techniques), it may be difficult to determine if the target's corners were rounded and/or to point to the target once identified. These issues are mitigated somewhat by the option to cross-validate the N_T results with the eye-tracking data (since eye-tracking data provides a close measure of where visual attention was directed), and by the fact that these potential confounds were less likely to have been an issue with the other highlighting techniques.

Assuming that there are no other methodological shortcomings and other confounding factors, then this result may be weak evidence that: "**people are terrible at judging how distracting different stimuli are**". If true, it would have significant implications for interface design processes and practices, as the suitability of highlighting techniques is currently based purely on the subjective judgement of a designer and/or a small number of their co-workers. It would also further support the argument that only "eyeballing" designs without the assistance of objective tooling support is a suboptimal way to work [140].

Further experiments are needed to understand the exact source of this mismatch between the two different results in our experiment.

7.4.5 Target cueing and alternative interpretations/uses for this protocol

To indicate which item is the target, a highlighting technique is applied to the target item. This means that there are actually two highlighting techniques present: the one being evaluated (A), and the technique being used to cue the target (B). The first implication is that care is needed when selecting the reference/cueing (B) technique used. Specifically, the experiment needs to be sufficiently sensitive to detect the differences between a wide range of different highlighting techniques. To achieve this goal, the cueing technique has been chosen to reduce the chance of false negatives (i.e. no significant distraction effects were observed because B was a stronger/more salient technique than A).

The second implication is that because two different highlighting techniques are displayed, the experiment method actually measures the interactions between the techniques used. That is, the distraction conditions measure which of the two techniques is stronger (i.e. if A is stronger than B, we should expect a greater amount of "distraction" or path divergence towards A in the distraction conditions), while the noticeability conditions investigate the effects of combining both (i.e. What effect if any does combining the two highlighting techniques have on user performance? How much more noticeable are items where both techniques are combined?).

Therefore, there are two ways that this method could be used to compare highlighting techniques.

1. **The first is to define one suitably "weak" technique as a reference point, and compare all other techniques in terms of the relative performance relative to that one**. If the usage of this "weak" technique becomes standardised, then it becomes relatively easy to develop a database of the relative effects of different highlighting techniques, and thus a model of their performance based on this database.
2. **The second is to use this method to study the the interactions between a given pair of techniques**. Specifically, it provides a way for testing the noticeability and distraction of A and B in isolation, but also the noticeability and distraction of the $A + B$ combination.

7.5 Conclusions

In conclusion, we successfully developed an experiment methodology for measuring the noticeability and distraction characteristics of highlighting techniques, and ran a study using this technique to quantify the effects of several commonly used highlighting techniques. *Task Time* and *Time to First Fixation* metrics were found to be useful measures of human performance.

One of the key findings of the study indicated that noticeability and distraction are not necessarily linked together by a strictly monotonically-increasing relationship; instead, it is possible to find highlighting techniques which are at least as noticeable but less distracting than a given technique. Another key finding was that humans have a poor ability to judge how noticeable and distracting a highlighting technique actually is.

Of the highlighting technique studied, `HPulse` was the most noticeable (but second-most distracting), `HColor` was perceived as being the most effective, while `LShootingStar` the worst overall. `HShake` was found to be the most noticeable technique if a low-distraction technique is required.

8 Study 2 – Animated Window Borders

8.1 Introduction

The previous chapter presented a method and validation study for measuring the noticeability and distraction effects of highlighting techniques when used in an abstract visual search task. This made it possible to empirically and objectively quantify how much the highlighting techniques helped or hindered the ability of participants to complete visual search tasks. However, visual search tasks are not the primary use case for highlighting techniques. Instead, highlighting techniques are often used for communicating that there is a notification that the user needs to attend to. Common examples of highlighting techniques used to implement notifications include the red-number badges on app icons indicating unread notifications, the bouncing dock icons on Mac OSX indicating that some action is in progress, or the way that the titlebar and taskbar buttons flash when trying to get the user's attention. In these cases, highlighting techniques are used to draw user attention away from their primary task (e.g. writing an essay, painting a picture, or filling out their tax returns) to focus on a secondary and unrelated activity (e.g. responding to an email or a poke on messaging app). Therefore, we wished to examine noticeability and distraction effects in a more ecologically valid context.

This chapter describes a study investigating the noticeability and distraction effects of highlighting techniques applied to window borders. Desktop/windowing environments provide a good for context measuring the effects of highlighting techniques for several reasons:

1. **Complex Multi-Tasking Environment** – Studies have found that users routinely have multiple windows open, and that these windows span a variety of different tasks [151]. For example, a typical workspace may concurrently have an email client, an instant messaging application, a web browser, a music player, and multiple work/task-related windows open. Examples of task-related windows include a word processor and multiple PDF viewers; a text editor, terminal, and version control client; or a graphic design application and image viewers displaying reference material. Applications need to compete with each other (and sometimes, other instances of themselves) to capture the user's attention in such multi-tasking environments.

2. **Familiar** – Windowing environments are widely used, and most participants should be quite familiar with their use and operation.

3. **Ripe for Innovation** – There are interesting opportunities for using highlighting techniques to improve the usability of the windowing environments. For example, highlighting techniques could be used for making it easier to quickly identify the active window, or for drawing user attention towards windows which require immediate attention. Inspired by Chang and Ungar's work [47], we also wanted to challenge the commonly expressed design guidelines to "avoid gratuitous animation" and to "avoid animating everything" [14]: with the widespread availability of modern animation-

friendly UI toolkits such as HTML5/CSS3/JS, QML/QtQuick [53], Android's Material Design [76], there are many unanswered questions about what can be done with the newly available capabilities, and how they should be best employed.

Therefore, we believe that there is compelling motivation to investigate the relative merits of highlighting techniques with window borders.

8.1.1 Key Objectives

There were three main (classes of) objectives for the research presented in this chapter. These objectives lead to contributions to our understanding of the *design space* of animated window borders, *methodological* contributions for measuring the effects of animated window border highlighting techniques, and *empirical* data (i.e. results) about the relative merits of the effects studied. They can be summarised as being:

1. **Examination of the Design Space of Animated Border Effects** – To perform an initial exploration of the design space for animated border effects, to identify promising and/or representative examples of techniques that may be useful/interesting to deploy in user interfaces.
2. **Ecologically-Valid Method for Measurement of Noticeability and Distraction** – To demonstrate and validate a method for comparing the noticeability and distraction effects of highlighting techniques applied to window borders.
3. **More Noticeable but Less Distracting** – To demonstrate that the comparative quality of animated border effects can be determined by showing that one effect is more noticeable but less distracting than another.

As in the previous study (Chapter 7), we were interested in understanding the tradeoffs between noticeability and distraction. Specifically, we wanted to show that it is possible to find a pair of highlighting techniques where A is more noticeable than B, while being less distracting (i.e. A is a better or "more effective" highlighting technique overall).

8.1.2 Overview of Approach

First, we conducted a design exploration to understand the capabilities of the UI toolkit, and to gain some insights about the effects of the different animated border effects and the ways in which they could be manipulated. From this exercise, we identified a set of representative techniques to use in a user study of their effects: some border types were chosen as examples of techniques which could plausibly be deployed in real user interfaces, while others were chosen more for their "shock/novelty" value, in the hope that they would provide valuable insights about different classes of effects.

We then conducted a dual-task user study to investigate the noticeability and distraction effects that applying dynamic highlighting techniques to window borders has on user performance. Participants were presented with a screen resembling a desktop: there was a window containing a simple mouse-movement game (dot-following) in the center of the screen, and a collection of 10 smaller windows in an elliptical arrangement around it (see Figure 8.1). Participants were instructed to identify the appearance of a highlighted window as quickly as possible by pressing the spacebar, while using the mouse to follow a dot moving around a wavy circular path. At the conclusion of the experiment, participants were asked to complete a series of short questions about their subjective experience of each border type.

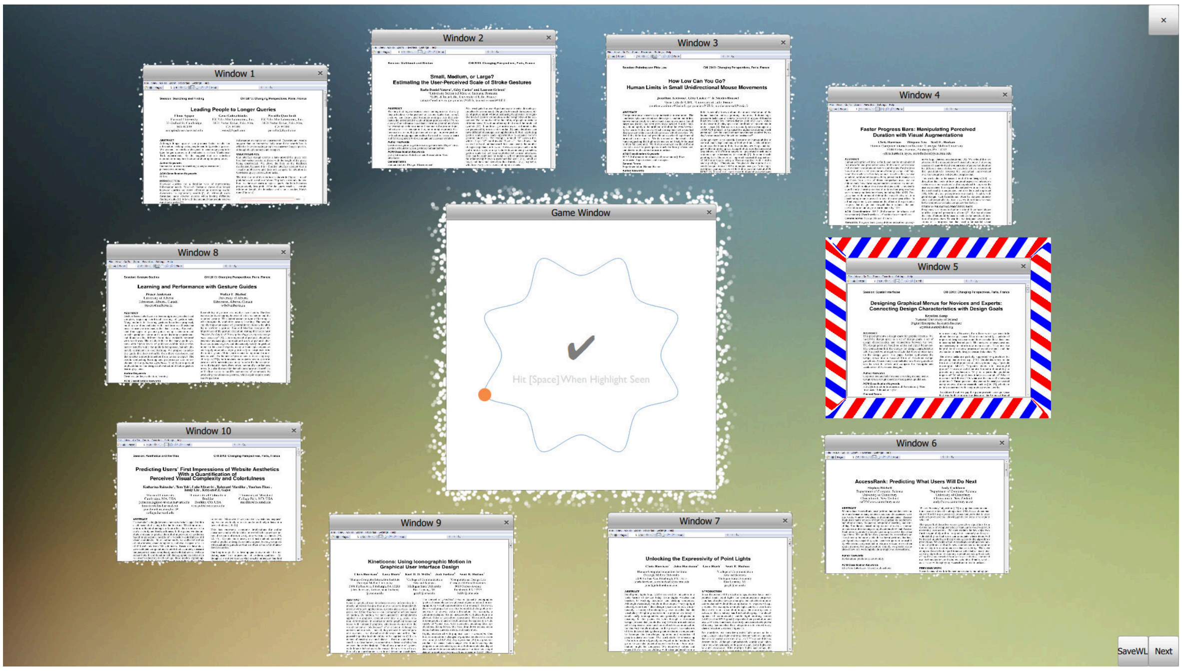


Figure 8.1: *Screenshot of the experiment setup during a trial. Participants are required to use the mouse to follow the moving orange dot on cog-shaped path, and tapping the spacebar as soon as they notice the highlighted window (i.e. Window 5, with the* `Barberpole` *border on it).*

8.2 Design Space for Animated Window Borders

This section presents the findings from our preliminary exploration of the design space for animated window borders (AWB's). We build upon the concepts introduced in our PCCH design framework (as presented in Chapter 4), demonstrating how the PCCH framework can be used (and extended) to inform and structure our understanding of a previously unexplored class of highlighting techniques. Our discussion of the design space for AWB's is divided into two parts:

1. **Construction** – First, we explore issues of how window borders can be constructed by combining different visual effects.

2. **Control** – Second, we explore the different ways in which the visual elements/effects identified can be manipulated to achieve different dynamic highlighting effects.

8.2.1 Construction of AWB's

According to PCCH, all "widgets" (i.e. buttons, controls, the cursor, and windows) can be described in terms of how they are constructed from layers of *Visual Elements* (see Section 4.3). Figure 8.2 shows how these concepts can be applied to create and/or describe a window with animated borders. The *Base* layer contains the *Visual Elements* for the animated border, while the *Contents* layer contains the window and its contents. Note that the origin point (shown using green dot and arrows) for the border is located inside the bordered-area and not on the outer edge – this makes it possible to change the border type without the window jumping around on screen (due to the changing border thickness). For the rest of this discussion, we are only focussed on what happens in the *Base* layer, where the border is defined.

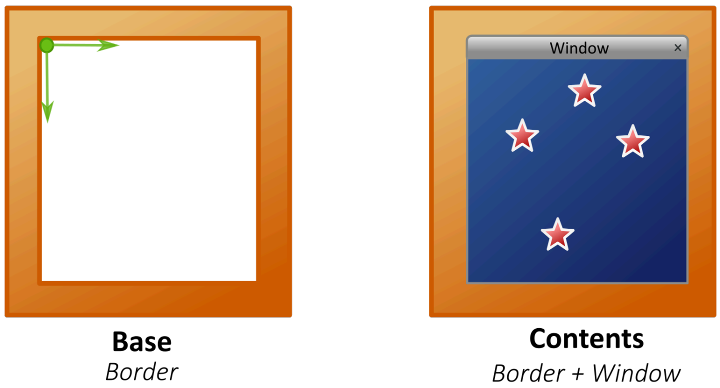


Figure 8.2: *Layers of visual elements to construct a window with an animated border. The border elements are shown in orange, and are placed in the "Base" layer.*

As mentioned earlier in Section 4.3.1, *Visual Elements* are "pixel-buffer generators" – that is, they describe how part of a widget is rendered as a two-dimensional array of pixels. The techniques (i.e. Visual Element types) used for constructing window borders fall into two main categories (with possibilities for hybrids), defined as follows:

1. **Particle Systems** (or "Particles") are a popular and well understood technique in computer graphics and game engines for creating visual phenomena where a region of space is filled in a semi-random fashion with a chaotic swarm of small discrete elements [134]. They can be used to simulate many different types of fuzzy physical phenomena including bubbles, splashes, swarms of animals/insects, specs of dust, falling rocks/debris, plumes of smoke, or even fire/fireworks.

   Different effects can be achieved using Particle Systems by varying the settings of the "*Emitter*" (i.e. the engine used to spawn and manage the lifecycle and behaviour of the thousands of *particles* used in the effect), and by using different "*Sprites*" (i.e. the images or objects used to represent/visualise each particle). By varying these combinations, different non-solid border effects can be created.

2. **Texture-Mapped Elements** (or "Textures") refer to effects where the colours of pixels in some region of space are individually manipulated using a shader to create some effect. (see Section 4.3.2 of the Design Framework)

   At the most basic level, all the pixels in a region can be set to a particular colour, or be made to copy the colours from some image buffer (via texture mapping techniques [147]). On a more advanced level, shaders (i.e. small snippets of custom code for graphics processing) can be used to procedurally generate patterns or to perform pixel-by-pixel manipulation as required [147].

   Most visual effects can be framed as examples of texture-based borders, by using the pixel-output of the visual effect as an image source for the border's texture-mapping shader.

Figure 8.3 provides an overview of how Particles and Textures can be used to create different types of window borders. Different AWB techniques can be created by taking a border construction method (i.e. either a *Particle System* based method, or one of the *Texture*-based techniques such as *Mosaic*, *Wrapped Border*, or *Background Panel* (BG Panel)), varying the contents of the source image/pixel-buffer passed to that border type (e.g. different shaped sprites, or different pattern-strips for the texture-based borders), and varying the settings/parameters of the border type used (e.g. adjusting the density/emission rate/velocity/etc. of the particle system, or the thickness of the borders).

8.2.1.1 Particle Sprites

One of the most critical parts of particle-based effects are the sprites used for representing the particles. We identified the following set of dimensions that can be manipulated to create different types of sprites as a starting point for further research:

- **Shape** – What shape is the object defined by the sprite? (e.g. Circle, Star, Bubble, or Butterfly)
- **Static versus Animated** – Is the sprite a static image (e.g. a plain circle), or is it a short looping animation clip (e.g. a butterfly flapping its wings)?
- **Uniform versus Varying/Combinations** – Is there only a single sprite used for all particles at the same time, or are there multiple sprites (including same-colour different shapes, same shape but different colours, or completely unique appearance)?
- **Size** – How big are the sprites? Large or small?
- **Filled or Unfilled** – Are the sprites solid/filled objects, or are they wireframes only?

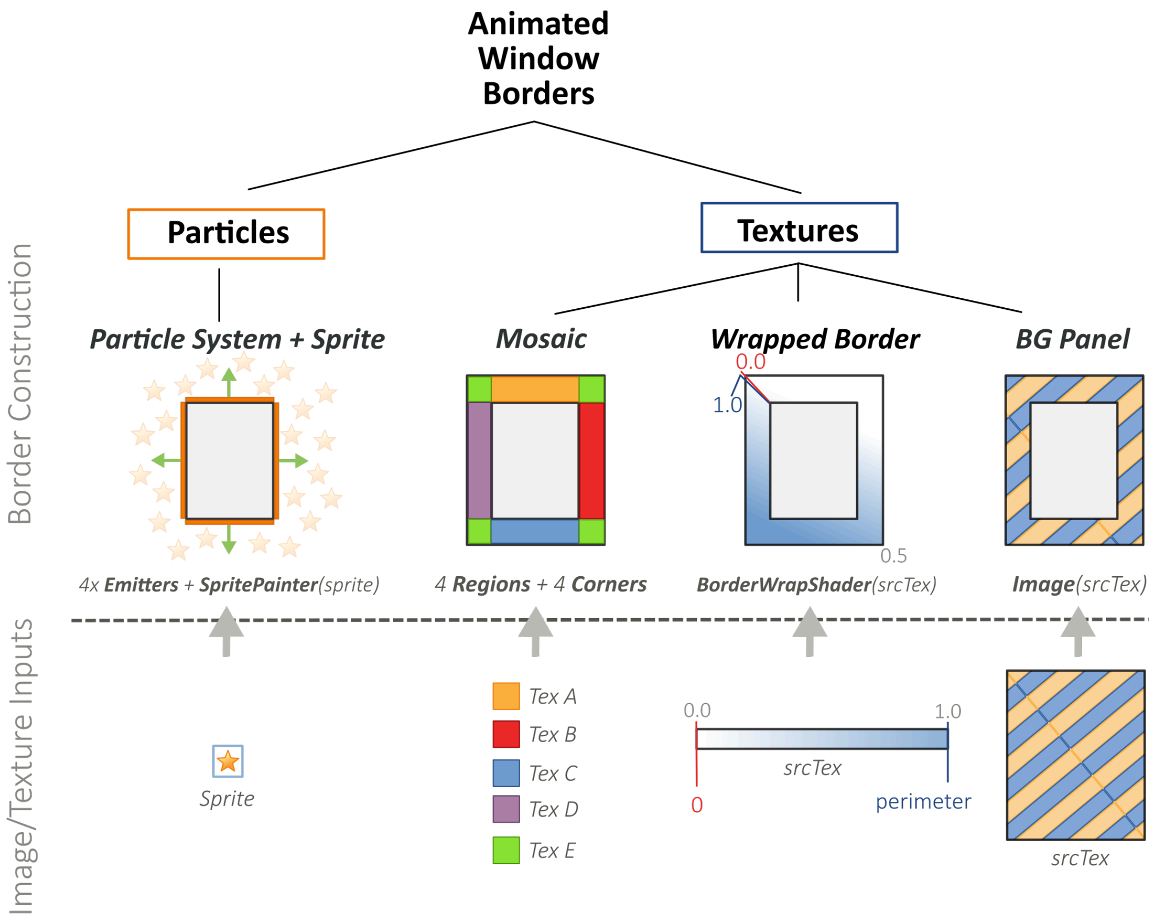


Figure 8.3: Overview/Taxonomy of border construction techniques. The top half shows how window borders can be constructed using Particle Systems or Textured Elements. The bottom half shows the image or pixel-buffer inputs that the border-construction methods shown above them use. Different window border types can be created by choosing different source images/pixel-buffers (bottom half) to pass to a border structure (top).

This set of dimensions does not represent the full scope and complexity of the design space; instead, it deliberately focusses only on the aspects that we were most interested in investigating, to simplify the discussion. In reality, each sprite can itself be described in terms of the full Highlighting Technique Design Framework (Chapter 4) as a tiny self-contained unit/component in a "Russian Doll"/Unix-pipes fashion.

8.2.1.2 Texture Types

There are three broad types of textures (as shown in Figure 8.4): **Procedurally Generated** (i.e. Shader Based), **Hand Crafted** (i.e. Raster/Vector Assets), or **Hybrid** (i.e. Procedurally generated, but using hand-crafted assets to control and populate the texture).

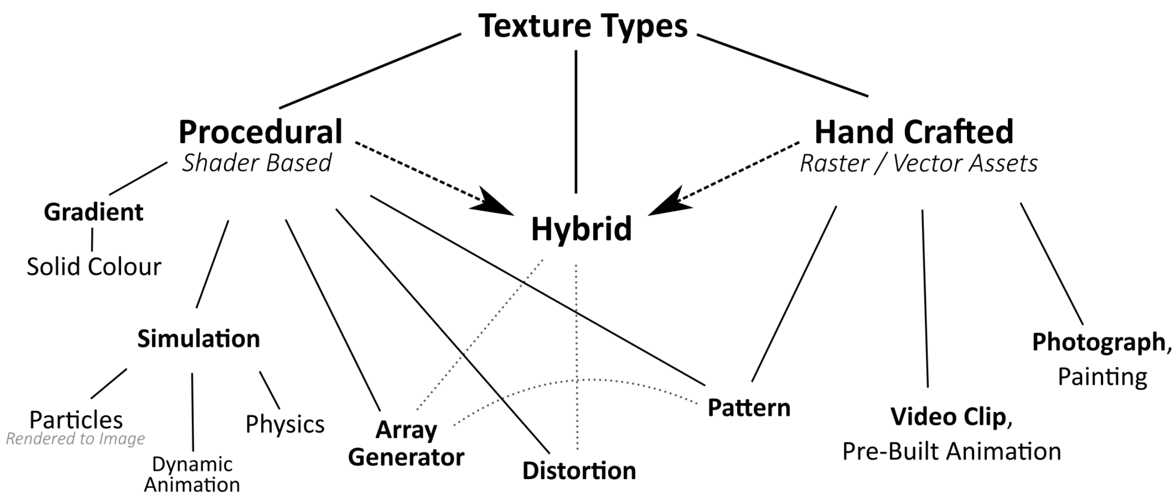


Figure 8.4: Overview of the different types of textures, and the ways in which they are related.

Procedurally Generated textures are more suited to geometrically simple but repetitive designs (i.e. stripes and other repeating patterns), or for complex dynamic effects which need to respond to user input (e.g. a border populated with simulated "pins and needles" (using a Particle System), where each pin dynamically pivots towards the cursor whenever it moves).

Hand Crafted textures are more suited to detailed/complex designs (e.g. photoreal rendered sprites) with a highly specific art style (and little to no dynamic behaviour), as these can be crafted by less-technical designers using WYSIWYG (What-You-See-Is-What-You-Get [94]).

Hybrid textures are created by combining hand-crafted assets with shader-based instancing and distortion of those elements.

8.2.2 Control of AWB's

This section presents a high-level overview of how the behaviour of the Particle System and Texture elements can be controlled to create animated window border effects. For further information about the specific manipulations that can be used for these elements (e.g. for controlling particle systems), consult the relevant sections of the QML documentation [53] or an equivalent toolkit.

8.2.2.1 Parameters of AWB's – Relationship to PCCH

The "*Particle System*" and "*Texture-Mapped*" Visual Elements introduced in Section 8.2.1 each expose a number of parameters that can be used to control their effects. For example, Particle Systems can be controlled by adjusting parameters such as the particle density, the size/shape of particles, and the speed at which the particles travel. These parameters can be animated using the PCCH animation-control structures outlined in Section 4.2.3, just like the `Scale` parameter was animated for `HPulse` in Study 1.

It is also possible to combine other manipulations discussed in the PCCH chapter (Chapter 4) with the Visual Elements introduced here for implementing AWB's. For example, interesting AWB effects can be achieved by applying animated colour-based manipulations – such as animating the Hue or Lightness of an AWB using a sine-wave F-Curve.

8.2.2.2 Direction of Movement

Border contents can be animated in one of the following ways:

- **In-Out** – The thickness of the border grows and/or shrinks over time. Examples of this include a constant stream of particles being emitted from the border and radiating outwards (e.g. the "`Fizzing`" borders technique, used in this study) and a pulsing glow (e.g. the "`Glow`" technique).
- **Around** – The contents of the border (i.e. a striped or repeating pattern) travels around the border frame. This creates an effect where the border contents appear to flow through a continuous tube wrapped around the window, pushed along by an invisible current. Examples of this include the `Barberpole` and `GreenStripes`.
- **In-Place** – For borders composed of multiple distinct elements (e.g. those based on particles), the border can be animated by applying a non-travelling effect to each element. That is, any visual manipulation can be performed on the particles, apart from animating their position/translation. Examples of this include the `BigButterfly` and `SmallButterfly` techniques (that use animated sprites of butterflies flapping their wings), or applying a `HPulse` or `LPulse` effect to each particle.

8.2.2.3 Nature of Movement

As discussed by Harrison et al. [81, 80], animated effects can be inspired by a wide range of sources. These include "*Nature-Inspired*" effects and those with "*Artifical/Man-Made*" origins.

- **Natural Motions** – Nature-Inspired effects are based on physical phenomena in the natural world. For example, the way that a stream of bubbles forming on the surface of an object placed in a carbonated beverage (`Fizzing`), or the way that a butterfly's wings flap when flying or when resting (`BigButterfly/SmallButterfly`).
- **Artificial/Man-Made** – Artificial effects are those with an anthropogenic origin. Examples include abrupt or "robotic" movements, such as the `Pulse` and `Shake` highlighting techniques from Study 7, or those based on traditional forms of advertising and signage (e.g. Barberpole and `FlashingLights`).

8.2.3 Example Techniques

Using the above framework, we constructed a set of example AWB techniques (described in Section 8.3.4). These techniques were chosen based on the one or more of the following criteria:

1. They are *commonly used the physical world* to draw attention to information/signage (e.g. `Barberpole`, `Flashing Lights`)
2. They are *promising candidates* for use in an "everything is animated" interface (e.g. `Fizzing`)
3. They are *easy to implement* (e.g. `Glow`), using functionality provided by the graphics toolkit used
4. They serve as a *contrasting example* for another technique (e.g. `GreenStripes` versus `Barberpole`, and `BigButterfly` versus `SmallButterfly`)

8.3 Method

This section describes how we conducted a dual-task study to empirically measure the noticeability and distraction of eight animated window border techniques. During the experiment, participants performed a dot following task using the mouse (i.e. they were instructed to follow the movements of an orange dot as it moved around a cog-shaped path) while tapping the spacebar whenever they detected a highlighted window within the mock desktop environment. Participants performed this pair of tasks bi-manually, with their dominant hand on the mouse and their other hand positioned over the keyboard/spacebar.

The primary task was designed to provide a continous measure of human performance from which "distraction" metrics could be computed. As in much of the prior literature, we used a dot following task (i.e. the participant uses some pointing device to track the movements of a moving target). This is because dot-following tasks provide a regular or near-continous stream of observable events arising from a tight *sensory-action motor control loop* [8].

We hypothesized that this tight coupling of hand-eye coordination can be used to detect of fluctations in mental load, as fluctuations in performance would arise from attention being directed away from the primary task to attend to the highlights (as participants were "distracted").

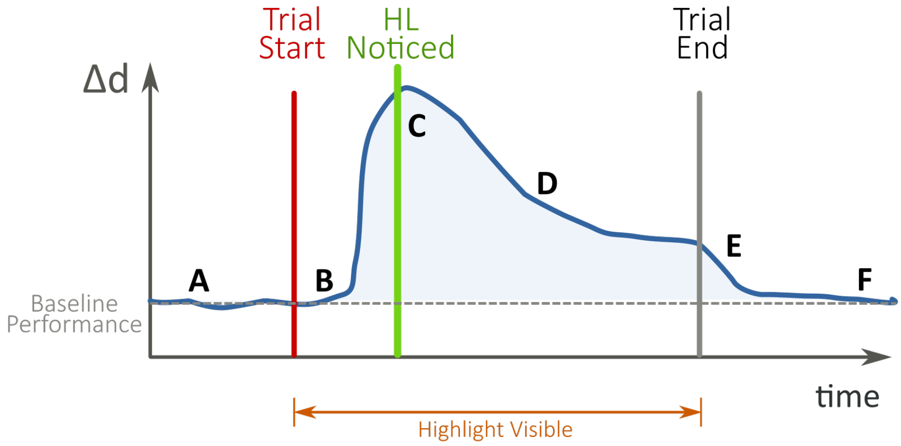


Figure 8.5: *Illustration showing the expected relationship between task performance and the presence of a highlight. The vertical axis shows the distance between target and cursor (Δd).*

Figure 8.5 shows a plot of the expected sequence of events before, during, and after each trial (where each trial corresponds to exposure to a highlighted window):

- **Section** ***A*** – There are no highlights present yet. Therefore, task performance from this region can be considered to be the "baseline" level.
- **Section** ***B*** – Although the highlight has appeared (at "*Trial Start*"), participants have not yet noticed the highlight (i.e. "*HL Noticed*"). Therefore, performance should not be affected yet, as they do not yet have conflicting demands on their attention.
- **Point** ***C*** – For a short period of time immediately before and after the "*Noticed*" event, there is a spike in Δd as participants became aware of the highlight, and redirected their attention towards attending to it.
- **Section** ***D*** **to** ***E*** – After attending to the highlight, participants would not have to do anything further with it, so they could try to ignore it. Therefore, there performance should begin to improve. However, since the highlight is still present and competing for the participant's attention, Δd should remain higher than baseline levels.
- **Point** ***E*** – At this point, the highlight has just disappeared (sometimes abruptly). However, there should still be a short delay before any difference in task performance is observed (due to the limits of human reaction times).
- **Section** ***F*** – Task performance should return to "baseline" levels now that highlights are no longer present.

Unlike studies in the prior literature, the dot-following task used here was formulated as a steering task [6]. To our understanding, traditional dot-following tasks [86, 28] feature a target moving in a unpredictable semi-random walk. However, pilot testing revealed an unacceptable level of noise in the data from traditional random-walk methods, due to the inherent unpredictability in those methods (Appendix A.2.1). In contrast, our formulation provides a much more controlled approach, while keeping the task challenging enough that participants have to stay engaged.

8.3.1 Hypotheses

As in the previous experiment, the main objective of this experiment was again to show that it is possible to rank the quality of highlighting techniques by measuring their effects on human performance. There were three groups of hypotheses: 1) Hypotheses related to the main objectives of this thesis, 2) Hypotheses about the validity and usefulness of the measures, and 3) Hypotheses about the effects of the border conditions studied.

8.3.1.1 Main Hypothesis – More Noticeable and Less Distracting

The first group of hypotheses concerned the main objectives of this thesis.

As in the previous experiment, the main hypothesis was that it is possible to find a pair of highlighting techniques where one is more noticeable but less distracting than the other (i.e. **H1.2**).

H 8.1
Across highlighting (border) techniques, measures of noticeability do not always increase monotonically with increasing distraction.

8.3.1.2 Validity and Usefulness of Metrics

The second group of hypotheses concerned the validity and usefulness of the metrics used for measuring noticeability and distraction. This is important because **H8.1** implicitly assumes that the dependent measures were valid, useful, and effective measures of noticeability and distraction. The following hypotheses concerned the validity and usefulness of the measures used in this experiment for measuring noticeability and distraction.

We expected that *Mean Displacement* (Δd_{avg}) would be the most suitable measure of distraction in the primary task, as it would account for all fluctuations in performance for the duration of a trial (as opposed to only using a single "extreme" point from each trial, as *Peak Displacement* (Δd_{max}) and *Minimum Displacement* (Δd_{min})).

H 8.2
Mean Displacement (Δd_{max}) is a sensitive and useful measure of distraction caused by the presence of highlighted windows during a dual-attention dot-following task.

Manipulation Checks:

- *When distracted, Mean Displacement should be higher than in the* `Baseline` *condition (where there should be no highlighted windows present, meaning that participants should not need to respond to anything).*
- *It should be possible to discriminate between different border types (i.e. there are significant differences between conditions).*

Conversely, we expected that *Peak Displacement* (Δd_{max}) and *Minimum Displacement* (Δd_{min}) would be less sensitive, as they were more likely to be affected by extreme outliers (due to these metrics using fewer datapoints):

H 8.3
Peak Displacement (Δd_{max}) should be similar to Δd_{avg}, since dips in performance (i.e. high Δd_{max} values) would be the result of participants being temporarily distracted (i.e. their attentions temporarily diverted away from the dot-following task). However, compared to Peak Displacement, this measure should have more noise/variability.

H 8.4
*Minimum Displacement (Δd_{min}) is the least sensitive of the distraction measures, as it should occur during the "before-onset" period (i.e. point **B** in Figure 8.5), when participants are still able to accurately track the dot with a high level of accuracy as they are not yet aware of any distractions.*

There was also the implicit assumption that the noticeability measure was valid and effective. Given that noticeability has been well studied, this should be somewhat redundant, but is included for completeness.

H 8.5
Time to First Noticed (t_n) is a sensitive measure of noticeability, such that it is able to discriminate between participant's response times to different highlighting techniques.

We were also interested in how closely the results from the performance metrics matched the subjective experience responses.

H 8.6
Rankings derived from noticeability and distraction analysis directly correspond with subjective experience responses.

8.3.1.3 Effects of Border Types

The third group of hypotheses concerned the border types being studied – for example, which ones would be more or less noticeable and/or distracting, and the relationships between different pairs of techniques.

"Naturally-inspired" highlighting techniques are less distracting and favoured more by participants than "Man-Made/Artificial" techniques:

H 8.7
"Naturally-inspired" techniques (e.g. `BigButterfly`, `Fizzing`, `SmallButterfly`, `SmallFizz`) should be less distracting and/or favoured more (i.e. they are more likeable/appealing/pleasing, or less annoying) than "Artificial/Man-Made" techniques (e.g. `Barberpole`, `FlashingLights`).

For each pair of similar techniques, the larger or higher strength version should be more noticeable but also more distracting:

H 8.8
`Barberpole`, `BigButterfly`, and `Fizzing` should be more noticeable and more distracting than `GreenStripes`, `SmallButterfly`, and `SmallFizz`, as the former techniques all use "higher strength" parameters.

`Barberpole` was designed to be the most noticeable and annoying/distracting, combining high contrast colours with a fast moving circular-travelling motion.

H 8.9
`Barberpole` should be the most noticeable and distracting technique.

8.3.2 Apparatus

Stimuli were displayed on a 23 inch TFT monitor (60 Hz refresh rate, 5 ms response time, white luminance 300 cd/m2 (rated)) running at 1920 x 1080 (HD) resolution, and fitted with a Tobii TX300 eye tracking device. The experiment was run on a Windows 7 workstation with an i7-3770 processor at 3.40GHz, 8 GB RAM, and Nvidia GeForce GTX 650. Experiments were conducted in a room lit primarily using standard office lighting (fluorescent tube/strip lighting overhead), and with natural light from a nearby window (filtered by fully closing blinds). Participants performed the experiment using a Microsoft "Wheel Mouse Optical" (`P/N X802382`) 3-button wired mouse. All handling characteristics were left at the operating system defaults. Participants were instructed to recenter the mouse before beginning each block so that they could comfortably complete the dot-following task without clutching or straining at any point along the path.

The experiment software was constructed using Python 3.4 (64-bit) using the PyQt5 toolkit (version 5.5.1 (64-bit)). The stimuli were implemented in QML, as the QML/QtQuick framework is a modern cross-platform technology designed for implementing dynamic user interfaces (complete with animated elements/transitions, particle effects, and complex shader-based effects) [53]. The experiment software was displayed full-screen (i.e. the titlebar and taskbar from the host operating system were not visible) to give participants an immersive mock-desktop environment.

Eye tracking and eye tracker calibration were performed using Tobii Studio 3.3.1. Participants were told to sit comfortably such that they could clearly see the screen and freely move their mouse-arm. To optimise tracking accuracy, participants were seated approximately 60cm away from the eye tracker unit. According to the Tobii SDK User Manual [5], eye tracking accuracy is highest at this distance. No physical restraints (e.g. chin rests) were employed to ensure that the participant's head position stayed constant; instead, participants were simply instructed to sit still. During the experiment, the "Track Status" window was displayed on a secondary monitor so that we could prompt participants to adjust their posture if tracking was lost.

8.3.3 Participants

We recruited 22 participants (14 male, 8 female) from a local university. Participation was voluntary, with participants aged between 20-60 (with a median age of 24.5 years); most participants were graduate students studying Computer Science. Data from two participants (1 male, 1 female) was excluded as these participants did not correctly perform the experiment procedure.

Participants had normal or corrected-to-normal eyesight. For precautionary reasons, participants were asked to not participate if they were susceptible to epileptic seizures. Eye tracking data was not available for several participants due to difficulties experienced during the calibration process for those participants.

Participants were asked the following demographic questions to account for potential sources of bias arising from their prior computing experience:

1. How many windows (on average) do you typically have open on your computer? [1]
2. How long do you spend playing games (e.g. mobile, console, PC, etc.) a week?
   0-5 hours, 5-10-hours, 10-20 hours, 20+ hours.

Participants had an average of 7.45 windows open (minimum 3, maximum 18, median 6), and spent an average of 2.5 hours playing games a week (median 0).

8.3.4 Stimuli

The experiment featured a field of 11 windows – a central "task" window, and 10 smaller "candidate" windows in an elliptical arrangement around it. During the experiment, participants performed a dot-following task in the central window, while looking out for when one of the candidate windows became highlighted. Participants were instructed to respond to highlighted window by tapping the spacebar as soon as they noticed the highlight.

[1] For this question, "windows" were defined to include applications that were open on the taskbar and/or in different workspaces (in order to account for multiple applications being run full screen). However, browser tabs were excluded, since these are not typically handled by the operating system's window manager.

Animated border effects were present on all windows at all times. In each trial, one of the candidate windows would get highlighted by having one of the following animated border conditions applied to it. At all other times, the non-highlighted candidate windows had a low-strength version [2] of the `Fizzing` effect applied to them. In addition, at all times, the task window always had the full-strength `Fizzing` effect applied.

8.3.4.1 Border Types

The following 8 animated border conditions were investigated in the experiment (see Figure 8.10 for examples of all these techniques):

- **Barberpole** – This effect is based on the traditional "Barberpole" pattern featuring a repeating pattern of red, white, and blue diagonal stripes. It is animated by making the striped pattern loop around the edges of the highlighted window at a slow-moderate speed.
- **Green Stripes** – This effect is similar to Barberpole, except that it uses two "calm" shades of green. There is low constrast between the the "light" and "dark" stripes.
- **Big Butterflies** – The edges of the highlighted window are covered in a swarm of medium-sized butterflies, making large flapping motions. Each butterfly is around 10-12mm in diameter. All butterflies are the same colour (i.e. magenta)
- **Small Butterflies** – This effect is similar to Big Butterflies, except the butterflies are much smaller (i.e. 5-6 mm in diameter) and have a more subtle flapping effect.
- **Fizzing** – Small white particles are emitted from the sides of the highlighted object. The effect is similar to that of a small tablet/pill dissolving in a glass of water, or bubbles forming around a straw sitting in a bottle of a carbonated beverage. The particles emitted are 8 pixels in diameter, and travel 2 cm in a dense cloud out from the edges of the windows.
- **Flashing Lights** – This effect imitates the look of the"flashing light" borders on old-style billboards, where a ring of lightbulbs flash in an alternating pattern to look like a line of dots is travelling in a loop around the border.
- **Glow** – A soft red glow is applied to the edges of the window. The glow grows and shrinks over 3-4 seconds.
- **Control** – No candidate windows recieve any additional highlighting effects. This condition is only to ensure that participants are performing the tasks correctly.

8.3.4.2 Primary Task Window

The primary task was conducted in a 440 $\times$ 440 px window with a background color of `#F0F0F0`. A detailed discussion of the primary task is presented in Section 8.3.6.1.

[2] Compared with the full-strength effect for the highlighted states, the low-intensity `Fizzing` effect on inactive candidate windows had smaller (5px) particles that travelled a shorter distance (i.e. 4mm)

8.3.4.3 Candidate Windows

Candidate windows were 300×200 px windows arranged in an elliptical pattern around the central (primary task) window. They were populated with scaled-down screenshots of CHI papers from the past 5-10 years, showing the top-half of the first page of each. Academic papers were chosen to provide a set of ecologically and contextually valid stimuli, while still retaining a relatively homogenous appearance to avoid making any windows particularly visually salient. It was assumed that academic papers are ecologically valid since it is often necessary to have multiple windows of papers open when searching/comparing these [151].

Each screenshot depicts a PDF viewer showing the top-half of a CHI full/short paper (typeset using the ACM conference proceedings template). No images or graphics were visible within the screenshot area to avoid having any easily distinguishable features. Screenshots were created by resizing PDF viewer windows to be 913×665 px, normalising the zoom levels of the document (e.g. "Fit to Window" and scrolling to the top of the file), and cropping the screenshot to only include the window contents (skipping the window chrome and surrounding windows). The resulting screenshots all had a similar grayscale appearance, with 1-2 lines of titles, followed by two columns of blurry gray text.

At the start of each block, the layout of the candidate windows was changed by randomly shuffling which window would display which screenshot. The locations of the windows never changed; they remained fixed in the elliptical layout, while the contents of each window would change instead. The same set of 10 screenshots would be used across all blocks. All participants used the same set of screenshots.

8.3.5 Design

The experiment analysis was structured as a 1×7 within-subjects ANOVA. There was a single factor, *Border Type* (or simply, *Type*):

$$\text{Border Type} \in \left\{ \begin{array}{l} \text{Barberpole } (bP), \text{Green Stripes } (gS), \\ \text{Big Butterfly } (BB), \text{Small Butterfly } (SB), \\ \text{Fizzing } (fz), \text{Flashing Lights } (FL), \\ \text{Glow } (GL) \end{array} \right\}$$

Participants performed 5 blocks of trials. In each block, they interacted with each of the 7 techniques once. Within each block, the order in which each condition appeared (along with the time delay between trials) was fully randomised. Participants were told that the first block of trials were for training purposes only; as a result, these results were collected but not included in the analysis.

There was also a *Control* condition that was included once per block. In this condition, none of the windows would get highlighted, but participants would still get notified that a window had been highlighted. This condition was designed to verify that participants were performing the task correctly, and that they were not simply responding to the onset notification. If participants were correctly performing the task, there should be no noticeability events generated for this border type. For this reason, the *Control* condition was not included in the Noticeability and Distraction analyses.

8.3.6 Tasks

Participants had two tasks during the experiment:

1. Perform a dot-following task
2. Respond when they detected and identified a highlighted window

This dual-task design was chosen to simulate the use of highlighting techniques to notify the user of a window requiring attention.

8.3.6.1 Primary Task – Dot-Follow

During each block, participants used a mouse to perform a dot-following task as shown in Figure 8.6. A round orange dot travelled around a cog-shaped path (shown as a thin light blue line). Participants were instructed to keep the cursor as close to the center of the dot as possible; when the distance from cursor to the center of the dot increased, the dot would grow larger (from a diameter of 10 px to 25 px) and its colour would change to a lighter yellow-orange shade to indicate that their performance had degraded.

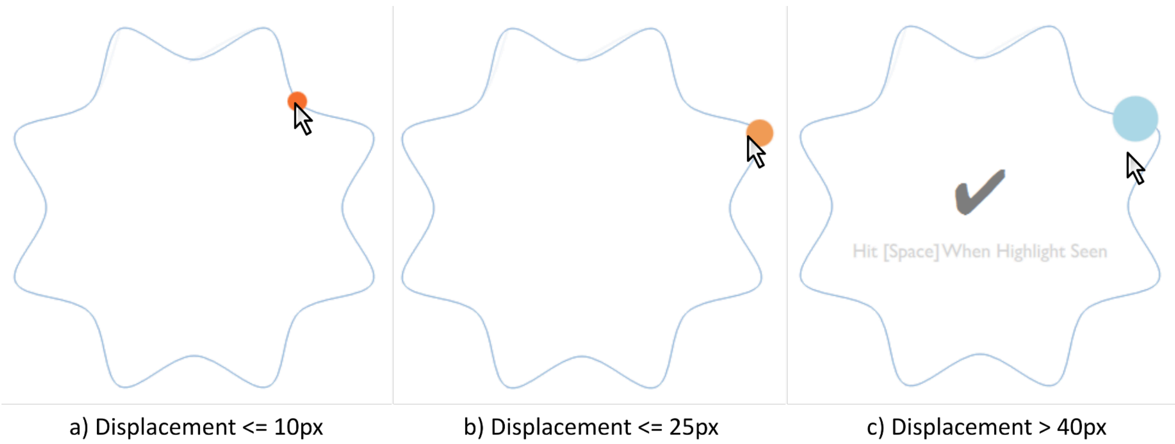


Figure 8.6: *Screenshots of the dot-following task. Participants were instructed to keep the cursor inside the coloured dot. The size and colour of the dot reflected how well participants were performing the task – a small orange-red dot (**a**) indicated good performance, a slightly lighter coloured and larger dot (**b**) indicated declining performance, and a large light-blue dot indicated terrible performance (c).*

Figure 8.7 shows the construction of the cog-shaped path. The cog had 16 points – 8 peaks and 8 troughs. The start point was located in the middle of a trough at the top of the cog (i.e. at the 0° marker). Peaks were 170 px from the center of the path, and troughs were 130 px from the center. Each complete revolution of the cog took 15 seconds to complete. During each block, the dot completed 13 revolutions of the cog. The path travelled by the dot was implemented using a Catmull-Rom spline [46].

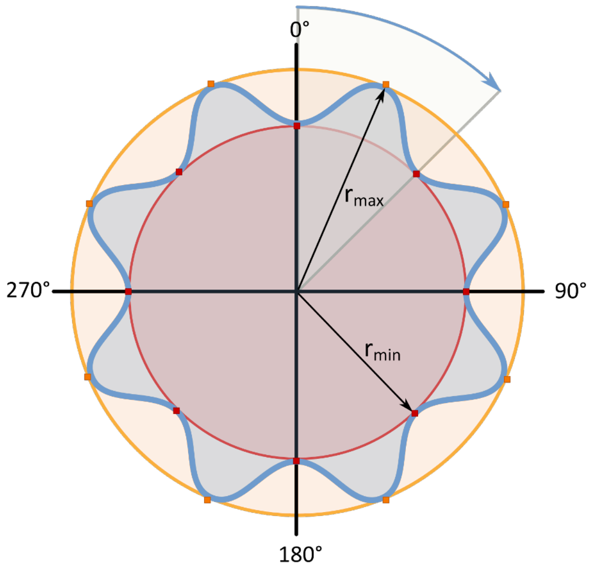


Figure 8.7: *Construction of the wavy circular path used for the primary task. The wavy dark-blue line is the path followed by the orange dot. Square-dots indicate the 16 points used to define the Catmull-Rom spline. The red-circle indicates the inner radius ($r_{min} = 130px$), and the orange circle the outer radius ($r_{max} = 170px$). The path was constructed by stepping clockwise around the circle.*

8.3.6.2 Secondary Task – Noticeability of Highlights

During each trial, one of the candidate windows would get highlighted using one of the border types. Participants were instructed to tap the spacebar as soon as they had detected which one of the windows had been highlighted. The spacebar was chosen as it is large and relatively easy to press button, eliminating the risk that participants would have to look away to find the key to press; participants were instructed beforehand to place and keep a finger on the spacebar to prevent the need to look. Mouse buttons were not used as we wanted to reduce the risk that participants would accidentally/involuntarily click while moving the mouse.

During each trial, an indicator appeared in the center of the cog-shaped path (as shown in Figure 8.6c) to indicate to participants that there was a highlighted window they should be attending to.

8.3.7 Dependent Measures

8.3.7.1 Time to First Noticed (t_n) Metric

The *Time to First Noticed* (t_n) metric measured how long it took for participants to notice the highlighted window after the animated border appeared. This was measured as the time taken for participants to press the spacebar after each trial had started. If the spacebar was pressed multiple times, only the first press was used. Lower values of this metric corresponded to more noticeable techniques, since the response time was lower.

8.3.7.2 Displacement (Δd) Metrics

The *Displacement* (Δd) metrics were used to measure task performance in the dot following task. On each mouse move event during each block of trials, the *cursor position* ((x, y)) and *distance-between-cursor-and-center-of-target* (d) were logged to a file. From this event stream, the three Δd metrics were computed by aggregating the d values in the following ways:

1. **Mean Displacement (Δd_{avg})** – The mean displacement was computed by averaging the Displacement values recorded during each trial, to compute an estimate of the overall performance, taking into account the peaks and troughs.
2. **Peak Displacement (Δd_{max})** – The peak displacement was computed by finding the highest Displacement value recorded during each trial.
3. **Minimum Displacement (Δd_{min})** – The minimum displacement was computed by finding the smallest Displacement value recorded during each trial.

We hypothesized that d values would be higher when participants were distracted, since the dot would have travelled further away from the expected position while they were distracted. Therefore, for all the border types studied, the Δd_{avg}, Δd_{max}, and/or Δd_{min} values should be higher than in the `Baseline` (i.e. when the participant is only performing the primary task) condition.

The `Baseline` condition used data from the "pauses" that occurred before, after, and between trials (see Figure 8.8), when no highlighting techniques were present (and participants were not cued to search for any either). Data from the first 3 seconds of each block (as determined from the pilot studies of the method, as per Appendix A) was discarded to account for the initial shock and subsequent delay before participants could begin accurately following the movement of the dot. Also, the first two seconds of each pause were also omitted to prevent the disappearance of highlights from skewing the results.

8.3.8 Subjective Experience Questions

In addition to the objective measures of performance, we also wanted gain insights on the subjective qualities of each technique. We were concerned with three main types of insights:

1. Percieved "Fitness for Purpose"
2. Generalised Likeability (or how well recieved the technique was)
3. Emotional Impact

This thesis is focussed on investigating how Noticeability and Distraction can be as metrics of the “Fitness for Purpose” (or effectiveness) of different highlighting techniques. Therefore, we were most interested in understanding how noticeable and distracting participants percieved the techniques to be. As shown in the first experiment, there can be significant differences between the percieved noticeability and distraction, and what objective task performance measures show. Therefore, we again asked participants to rank the techniques with respect to each other in terms of noticeability and distraction, from “best” to “worst” (i.e. most noticeable to least noticeable, or least distracting to most distracting). The qustions were worded as follows (emphasis added here for clarity):

1. Please rank the techniques in order of how **noticeable** (i.e. the ease of spotting or detecting) the effect is, from most noticeable (left-most) to least noticeable (right-most)
2. Please rank the techniques in order of how **distracting** the effect is, from least distracting (left-most) to most distracting (right-most).
   (Distraction may include annoyance, difficulty focussing on what you're doing, and/or visual disruption)

We were also interested in gaining a preliminary understanding of the emotional effects of these techniques and/or to understand how well recieved they were. Given that this experiment is focussed on achieving a higher-level of ecological validity than our first experiment, it is important to gain a broader picture about issues which may help or hinder the deployment of these techniques in more realistic usage scenarios. For example, we were interesting in understanding if (and how) the user's preference for/against a particular technique was correlated with how noticeable and distracting it was observed or percieved to be. We were also interested in understanding what may have contributed to participants to consider a technique to be more distracting than another: For instance, did they consider the effect hard to detect (forcing them to have to spend more effort hunting for it), annoying, or did the presence of the effect make it harder to focus on their primary task?

For each border effect, participants were presented with a series of 7-point Likert Scales. For each question, participants were instructed to provide a score (from 1-7) based on extent to which they agree/disagreed with each statement (1 = strongly disagree, 7 = strongly agree)

1. I **like** this effect
2. This effect is **visually attractive**
3. This effect is **pleasing/satisfying**
4. It was **hard to detect** the highlighted window
5. This effect is **annoying**
6. It was **hard to focus** on the moving dot when this effect was present

Finally, we provided participants with the opportunity to leave freeform comments about any of the border types they wished to.

8.3.9 Noticeability and Distraction Metrics

This section describes how noticeability and distraction metrics were calculated from the experiment data and from the subjective experience questions.

8.3.9.1 Performance-Based Metrics – N_t and $||\Delta d||$

The performance-based noticeability metric (N_t) was computed using the following equation from the *Time to First Noticed* (N_t) data:

$$N_t = (15 - N_t)/15 \tag{8.1}$$

This formulation was used as trials lasted 15 seconds, and lower response times (i.e. faster detection of the highlighted window) meant that a technique was more noticeable.

The performance-based distraction metric ($||\Delta d||$) was computed using the following equation from the *Mean Displacement* (Δd_{avg}) data:

$$||\Delta d|| = \Delta d_{avg}/50 \tag{8.2}$$

Mean Displacement was used as we expected that this would be the most appropriate metric of task performance. These values were divided by 50 to normalise the values (within the 0-1 range); participants were unlikely to exceed a displacement of 50 pixels, as they would receive strong visual cues that they had strayed too far already.

8.3.9.2 Subjective Ranking Metrics – N_s and D_s

The subjective noticeability and distraction metrics (N_s and D_s respectively) were calculated by reversing the order of the responses (since the most noticeable/distracting technique had the *lowest* rank of *0*), and dividing by the number of techniques (*8*). That is,

$$s' = (8 - s)/8 \tag{8.3}$$

8.3.10 Procedure

The experiment had five phases: Introduction, Calibration, Training, Data Collection, and Post-Experiment Survey.

8.3.10.1 Introduction Phase

Participants were welcomed and seated in front of the eye tracker. On screen, a demo version of the experiment software was running; this would be used to explain the procedure.

Participants were told that they were going to be playing a game where they were required to follow a moving dot with the mouse, keeping the cursor as close to the middle of the dot as they could. They were given the following instructions while having the procedure demonstrated:

> “ *Your task today is to follow the moving dot with the cursor, trying to keep the dot as small as possible. While performing this task, one of the windows the surrounding windows will get highlighted. Tap the spacebar as soon as you notice that one of these is highlighted. A "giant tick" will also appear in the middle of the screen whenever a window is highlighted. Just tap the spacebar anytime you think you think you've noticed a highlighted window.*

Once participants understood the procedure, they were asked to fill out the Informed Consent form (see Appendix C) along with the demographic questions noted earlier.

8.3.10.2 Calibration Phase

The eye tracker was calibrated using the same procedure used in the first study (see Section 7.2.7.2 for more details).

8.3.10.3 Training Phase

After calibrating the eyetracker and before starting the experiment, participants were shown a series of two screens clarifying the details of the experiment.

The first screen repeated the instructions for the experiment, and instructed participants to place their primary hand on the mouse, and their secondary hand over the spacebar (ready to tap the spacebar).

The second screen resembled the experiment task layout, but with all the border types used in the experiment displayed on the candidate windows. They were told:

> “ *Here are all the highlighting techniques you will encounter in today's experiment. Take note that there are ones with big dots and others with small dots. You need to respond to everything except the ones with small dots.*

Participants then performed one block of trials (see following section for the detailed procedure). They were told that this first block was just a practice run. During this first block of trials, we clarified the procedure if participants were having difficulty the correct procedure and/or answered any other queries participants may have had.

8.3.10.4 Data Collection Phase

During the Data Collection phase, participants performed a series of 4 blocks of trials. Between blocks, participants were encouraged to take breaks to relax their hands and eyes.

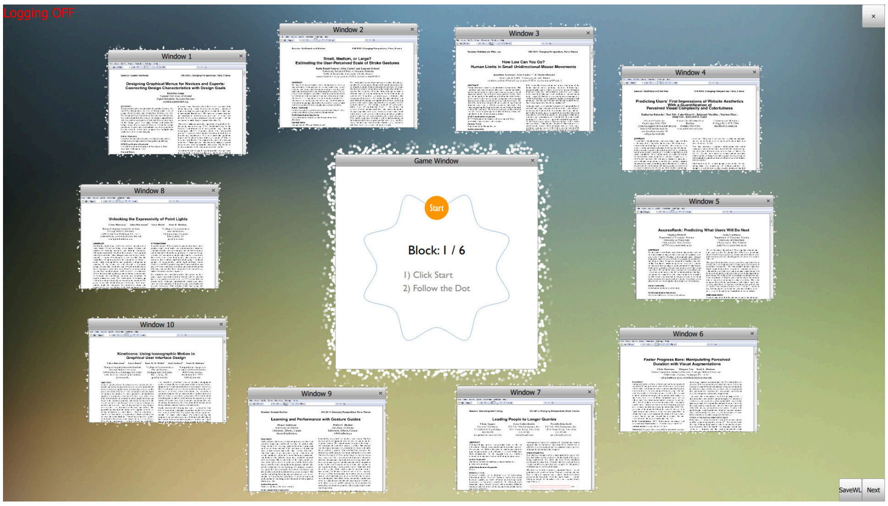
Figure 8.8 shows the structure of each block. During each block, participants would perform 8 trials – one per border condition (including the `Control` condition), with the conditions presented in a randomised order. First, there would be a few seconds for the participant to settle into the routine of performing the dot following task. There would then be a series of fixed-length trials and variable-length pauses between those trials.

Figure 8.8: *This shows how fixed-length trials and pauses between trials were interleaved.*

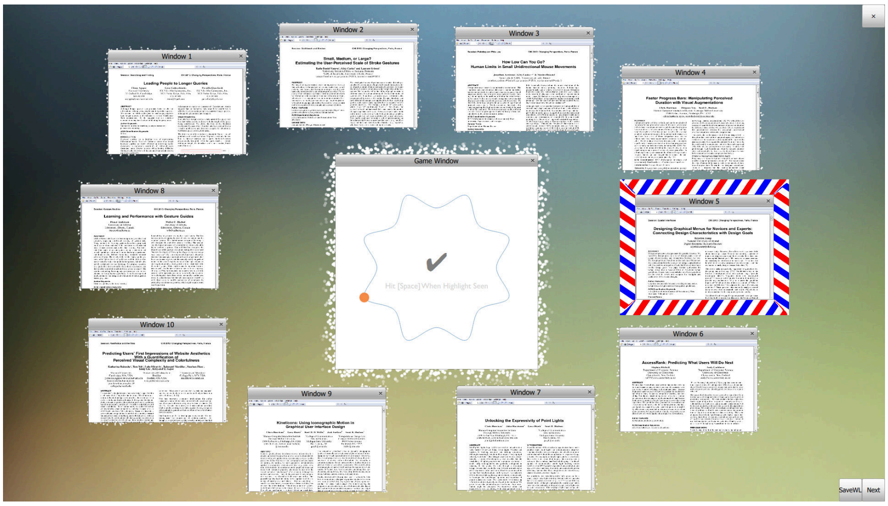
During each trial, one of the 10 candidate windows would get highlighted, and participants would have to hit the spacebar to when they had detected the highlighted window (see Figure 8.9(b)). Trials were all the same length so that we could investigate if there were any long-term disruptive effects to the highlighting techniques (e.g. perhaps one of the techniques would be quickly noticed, forgotten about, and then noticed again; or maybe, another technique causes a constant/steady stream of short saccades to keep checking on it). However, the pauses between these exposures would be of variable length, to make it less predictable to participants when the next trial may occur, making it harder to develop a rhythm.

Participants started each block of trials by clicking on a large round "Start" button placed where the orange dot would start (see Figure 8.9(a)). After the button was clicked, there was a 5-second "count-down", where the orange dot would flash several times to prime participants to be ready to start moving.

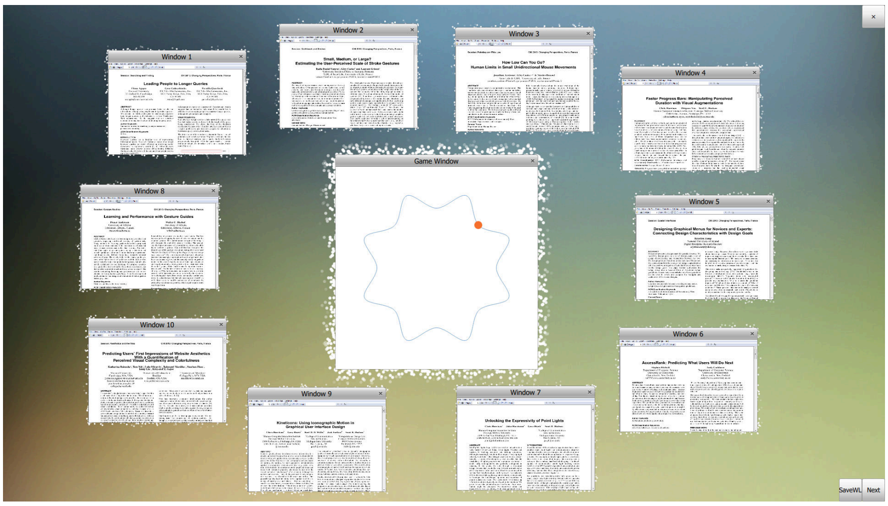
Before starting the trials, participants were verbally instructed to recenter the mouse and



(a) Before Start



(b) During Trials



(c) During Pauses

Figure 8.9: *Screenshots showing different stages in the procedure for each block.*

ensure that they could comfortably trace the outline of the cog-shaped path.

8.3.10.5 Post-Experiment Survey

Following the *Data Collection* phase, participants completed a post-experiment survey to collect data about their subjective experience of the highlighting techniques. This survey was also conducted using an interactive interface built from the same components that the experiment was implemented with; this made it possible to interactively display all the border types used in the experiment, exactly as they appeared, while participants were answering the survey. Participants were then shown a series of 4 screens.

On the first two screens (Figure 8.10), participants were asked to rank the border types in terms of how noticeable and distracting they were. This was done by dragging the windows for each border around so that the order (from left to right) reflected the desired ranking (from most noticeable/distracting to least).

Figure 8.10: *Screenshot of the subjective ranking tasks where participants were asked to rank the border types in order of decreasing noticeability/distraction be rearranging the borders. Here the borders are shown in order of decreasing noticeability. The distraction ranking question had similar presentation (except with a different prompt and labels).*

On the third screen (Figure 8.11), participants were then shown a set of 8 windows, one per border condition, with a short survey displayed inside each window. Participants were asked to fill out the questions in each window corresponding to the border for that window.

On the final screen (Figure 8.12), participants were again shown a set of 8 windows. As in the previous step, there was one per border condition. The borders were presented in the same order/places as on the previous screen, to minimise confusion about which technique they were responding to. However, this time participants were instead asked to provide comments on any of the border types by leaving a comment about that technique.

Figure 8.11: *Screenshot of the third part of the survey, where participants were asked to respond to a series of Likert-scale questions statements about each border type.*

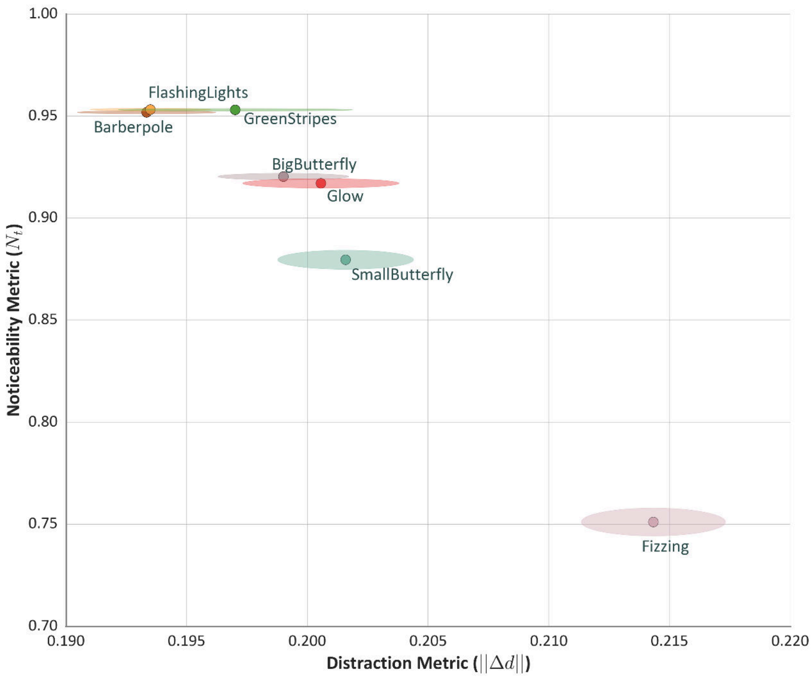
Figure 8.12: *Screenshot of the final part of the survey, where participants were asked to comment on any of the border types that they wished to.*

8.4 Results

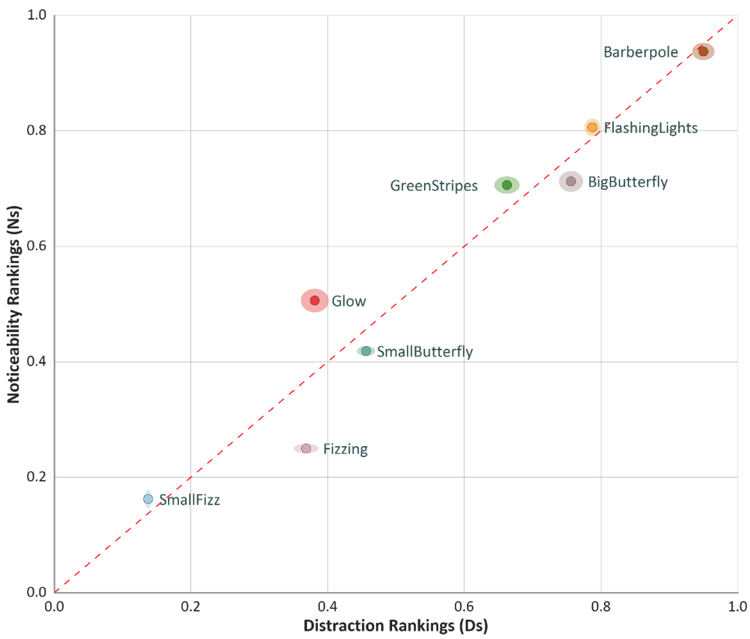
This section presents the results of the experiment described in Section 8.3. Unless noted otherwise, error bars on plots show ±1 standard error.

8.4.1 Noticeability versus Distraction

Figure 8.13 compares the relationship between Noticeability and Distraction for the different border types. Figure 8.13(a) uses metrics N_t and $||\Delta d||$ computed from the performance metrics using equations 8.1 and 8.2. Figure 8.13(b) uses metrics Ns and Ds, which were computed using equation 8.3, from participant responses to the subjective experience questions asking them to rank the border types. In both figures, techniques closer to the top left corner should be better than those in the bottom right corner (i.e. they are more noticeable and less distracting). The shaded regions around each point show ±1 standard error.



(a) **Task Performance** - N_t and $||\Delta d||$. Note the differences in scale and start/end points for each axis.



(b) **Subjective Ranking** - N_s and D_s. There is a strong positive correlation between N_s and D_s (Spearman $\rho = 1.0$), with most of the techniques falling close to (if not on) the 1-1 line.

Figure 8.13: *Comparison of measures of Noticeability and Distraction, using performance-based metrics (top) and subjective experience measures (bottom). Techniques closer to the top-left corner should theoretically be "better" (i.e. as they are more noticeable and less distracting). Shaded regions show ±1 standard error.*

As can be seen from Figure 8.13 the Task Performance and Subjective Ranking graphs appear to be horizontally-flipped images of each other: the Subjective Rankings lie along the $(0,0) \to (1,1)$ line, while the Task Performance datapoints lie along a line approximately parallel to $(0,1) \to (1,0)$. However, the relative order of the techniques is similar, with `Fizzing` at one extreme, and `Barberpole` / `FlashingLights` at the other. This can be seen more clearly by comparing Figure 8.14 and Figure 8.13(b).

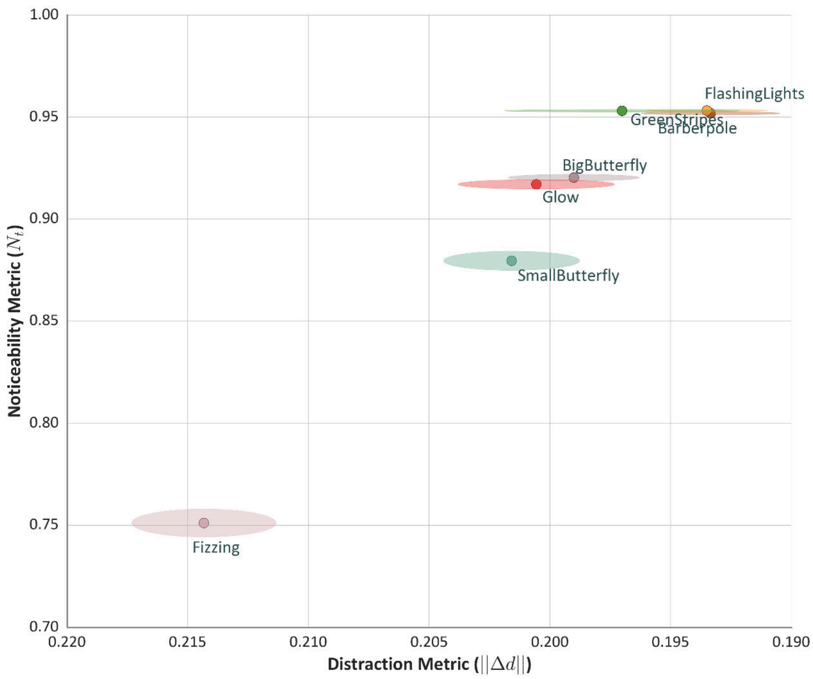


Figure 8.14: *A copy of the N_t versus $||\Delta d||$ plot (Figure 8.13(a)) with the horizontal axis inverted. Note also the differences in scale and start/end points for each axis.*

From these graphs, we can conclude that the Mean Displacement (Δd) metric does not appear to be a suitable metric for measuring "Distraction" (in the conventional sense), as it is the complete opposite of what should be expected given the subjective experience ranking data. While it is tempting to invert the horizontal axis (or to use the inverse values – i.e. $D_{\Delta d} = 1 - \Delta d$) as illustrated in Figure 8.14, doing so would result in the `Baseline` having a higher value than all the border types (i.e. it would be "more distracting" than being distracted by one of the highlighted windows). Furthermore, it is not clear how such a manipulation could be justified in terms of what it physically represents. The $||\Delta d||$ metric used here (i.e. $||\Delta d|| = \Delta d_{avg}/50$) arose from our initial hypothesis that when distracted, participants would not be able to perform the dot following task as accurately as when they were not distracted, and hence, the distance between the cursor and the center of the dot would be larger when participants are distracted (i.e. Δd and $||\Delta d||$ should be *higher* when performance is *more affected*, and *lower* when performance is *less affected*). However, an inverted version of this metric no longer retains this link.

8.4.2 Time to First Noticed (t_n)

Figure 8.15 shows a plot comparing the average time when each border type was first detected by participants.

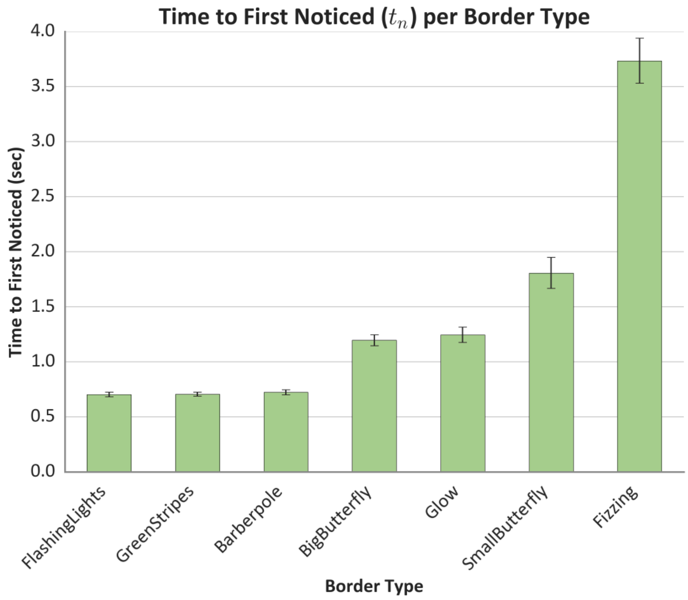


Figure 8.15: *Plot comparing the average time needed for participants to first notice each border type (by pressing the spacebar). Lower times are better (i.e. more noticeable).*

Four main groups or "levels" of noticeability were detected:

1. **High** – `FlashingLights`, `GreenStripes`, `Barberpole`
2. **Medium** – `BigButterfly`, `Glow`
3. **Low** – `SmallButterfly`
4. **Very Low** – `Fizzing`

We analysed the noticeability data as a one-way within-subjects ANOVA with 7 levels, and found significant differences between conditions ($F_{6,114} = 108.312, p = 0.000$). A post-hoc Tukey HSD test confirmed these groupings. This shows that this method was sensitive enough to detect some differences between different border types. However, it also reveals that this method was not sensitive enough to detect differences between similar techniques – notably, at the "High" noticeability level, this metric was unable to distinguish between the three techniques, with `FlashingLights` instead of `Barberpole` being the most noticeable technique (despite participants ranking `Barberpole` as being more noticeable), though the differences between these are all within the margin of error associated with those measurements. From the Noticeability versus Distraction plots earlier (e.g. Figure 8.13), it can be seen that the measurements for these three techniques were affected by a *ceiling effect* [167]. That is, the measure could not distinguish between these techniques as any differences were limited by the speed at which participants could react using their hands.

8.4.3 Displacement Metric (Δd)

The Displacement metric(s) measured how well participants could perform the dot following task in terms of distance between the cursor and the center of the dot (in pixels), and were intended as performance-based metrics of how distracting the border types were. This section examines three related metrics computed from this data: Mean Displacement (Δd_{avg}), Peak Displacement (Δd_{max}), and Minimum Displacement (Δd_{min}). We also present a section of the raw data (from which these metrics were computed) to contextualise these metrics.

8.4.3.1 Mean Displacement (Δd_{avg})

Figure 8.16 shows a plot comparing the mean displacement values for each border type. As expected, in the `Baseline` condition, Δd_{avg} was lower than in all the border types, as participants did not have any highlighting techniques to distract them. In contrast, in the control condition – `Absent` (where no windows were highlighted, but participants were cued to expect a highlight) – Δd_{avg} was higher than all the different border types studied. This could be because participants would start examining each window if they could not easily identify a highlighted window that was "supposed" to be present.

Surprisingly, Δd_{avg} results for the other border types were not as we had anticipated. As shown in Figure 8.16, `Barberpole` had the lowest value, `FlashingLights` the second lowest, `SmallButterfly` the second highest value, and `Fizzing` the highest value. However, `Barberpole` and `FlashingLights` were ranked by participants as being the most distracting techniques, while `SmallButterfly` and `Fizzing` (i.e. the ones with the highest Δd_{avg} values) were ranked as being the least distracting.

As already discussed in Section 8.4.1, these surprising results indicate that the Δd_{avg} metric was not measuring what it had been intended to measure. Instead, the Δd_{avg} metric appeared to be more like a secondary metric for noticeability. We were expecting that the Δd_{avg} values would measure "distraction" by showing that as participant attention would continually be drawn towards highly salient highlighting effects, those effects would have higher Δd_{avg} values than less salient effects. However, the opposite happened: less noticeable techniques, such as `Fizzing` and the control condition `Absent` (which simulated a pathologically unnoticeable effect) ended up having the largest Δd_{avg} values, while the most noticeable techniques (e.g. `Barberpole`, `FlashingLights`, `GreenStripes`) all had relatively small Δd_{avg} values. During the experiment, we observed on multiple occasions (e.g. most noticeably with ***p9***) that during the `Absent` conditions (and to a less extent, the `Fizzing` conditions), participants were frequently scanning over the entire screen hunting for a highlighted window, while such behaviours were relatively rare with highly noticeable techniques like `Barberpole`. Therefore, the obvious conclusion here is that Δd_{avg} was low on highly noticeable techniques, while it was high on barely-noticeable techniques, making it an "inverse noticeability" measure.

Closer inspection of the data shows that Δd_{avg} corresponds with the subjective experience responses to **Question 4** ("It was hard to detect the highlighted window") from the survey (Figure 8.21). For example, in both cases `Barberpole` had the lowest score (i.e. participants "Strongly Disagreed" with the statement that it was the hardest to detect, meaning it was in fact the *easiest to detect*), while `Fizzing` had the highest score (i.e. participants "Strongly Agreed" with the statement that it was the hardest to detect). This is further evidence that it was actually a measure of "inverse noticeability".

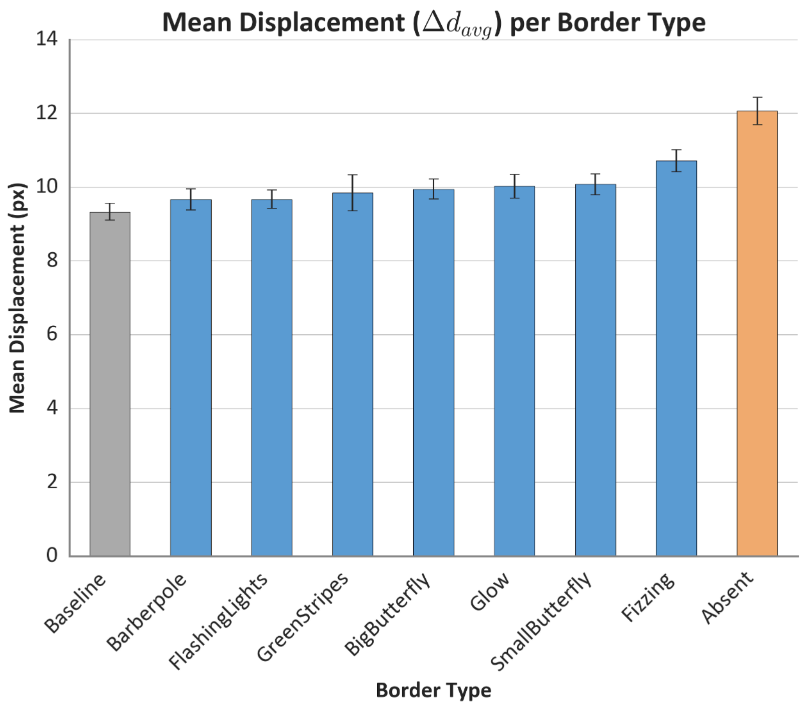


Figure 8.16: *Plot comparing the mean displacement (Δd_{avg}) for each border type.* `Baseline` *represents the Δd_{avg} during pauses (when no highlights were present).* `Absent` *represents the control conditions (with no window highlighted, but a cue was present). Note how the ordering of conditions is the opposite of the the Subjective Rankings of distraction (Figure 8.20(b).*

We compared the Δd_{avg} values as a one-way within-subjects ANOVA with 9 levels, and found that there was a significant difference between conditions ($F_{8,152} = 11.303, p = 0.000$). A post-hoc Tukey HSD test revealed that `Absent` was significantly different to every condition except `Fizzing`; however, it did not reveal any significant differences between the other conditions.

Given that the `Absent` technique appeared to be significantly different from the other conditions by visual inspection of the graph, we repeated this analysis with the `Absent` technique's data removed. The resulting one-way within-subjects ANOVA with 8 levels revealed that there was again a significant difference between conditions ($F_{7,133} = 3.796, p = 0.001$), with the post-hoc Tukey HSD test showing that there was a significant difference between the `Baseline` and `Fizzing` conditions.

These statistical tests show that although Δd_{avg} could discriminate between a few broad categories of animated window borders (i.e. completely-absent/barely-visible, and absent-but-notified), it was not sensitive enough to detect statistically significant differences between most types of borders studied.

8.4.3.2 Peak Displacement (Δd_{max})

Figure 8.17 shows a plot comparing the maximum or "peak" displacement values (Δd_{max}) for each border type. Overall Δd_{max} values were all similar to each other, with only the `Fizzing`, `Baseline`, and `Absent` techniques having slightly higher values. This suggests that the worst-case task performance in the dot-following task was relatively insensitive to differences between highlighting techniques. That is, for most of the highlighting types studied, participants did not experience any significant interruptions causing them to drift a long distance from the target; instead, any interruptions that occurred were likely to have been short or recurrent, allowing for relatively small but still variant mean displacement values. Conditions with higher Δd_{max} may have been the result of participants clutching during a trial – this could explain why `Baseline` has relatively high Δd_{max} values (e.g. some of the participants often waited until a pause to clutch/adjust the mouse position for comfort), and why most of the other techniques have similar values (i.e. in a few rare cases, some participants clutched during trials when they got uncomfortable).

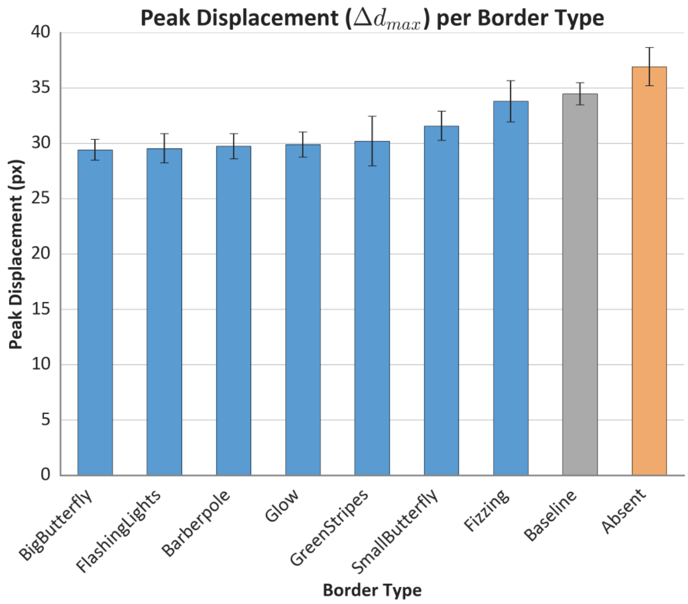


Figure 8.17: *Plot comparing the peak displacement (Δd_{max}) for each border type. Note the relative similarities of the values, and how* `Baseline` *does not have the lowest value.*

We analysed the Δd_{max} values as a one-way within-subjects ANOVA with 9 levels, and found that there was a significant different between conditions ($F_{8,152} = 4.486, p = 0.000$). A post-hoc Tukey HSD test found significant differences between `Absent` and `Barberpole`, `BigButterfly`, `FlashingLights`, `Glow`, and `GreenStripes` only.

8.4.3.3 Minimum Displacement(Δd_{min})

Figure 8.18 shows a plot comparing the minimum displacement values (Δd_{min}) for each border type. Surprisingly, the values and relative ordering of border types with this measure are closer to the results we had originally expected to obtain using the Δd_{avg} metric. For instance, `BigButterfly` (i.e. the third most distracting technique, as rated by participants) appears as the most distracting here (behind `Absent`), while `Barberpole` and `GreenStripes` (two of the most distracting techniques) feature in the middle-high end of the range, instead of having the lowest values. `FlashingLights` however is only ranked the second lowest value here, which is an interesting result. Another point of interest is how the `Baseline` and `Absent` conditions retain the same relative places as they did in the mean displacement (Δd_{avg}) rankings, but with a more significant effect size (e.g. `Baseline` appears significantly lower than all of the border types).

A possible interpretation for these results is that the minimum displacement metric represents "the best efforts of participants to follow the dot, despite the distractions". Using such an interpretation, higher Δd_{min} values would correspond to participants being less able to achieve high performance (i.e. perhaps it could be claimed that they were "more distracted"?) due to the stimuli present, whereas lower Δd_{min} values would correspond to participants being less affected by such distractions.

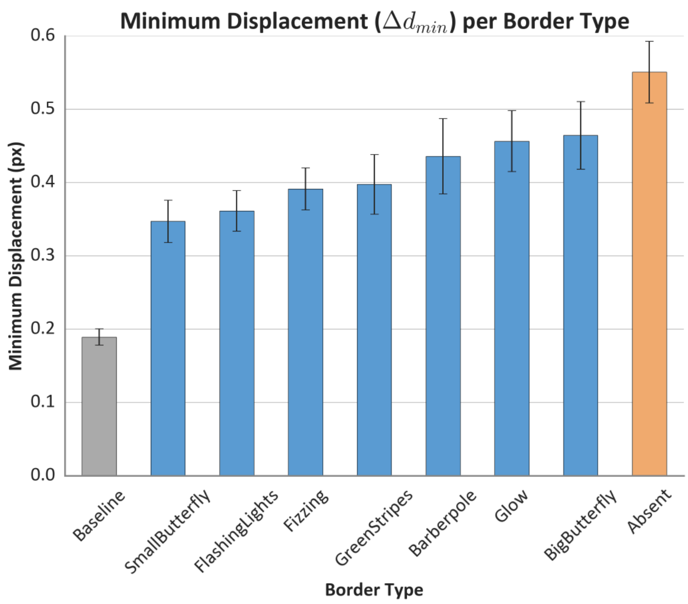


Figure 8.18: *Plot comparing the minimum displacement (Δd_{min}) for each border type. Note the increased similarities with the Subjective Distraction Rankings, while retaining Average Distraction characteristics (e.g. for `Baseline` and `Absent`). Also note the units/scale on the y-axis (which are less than 1 pixel).*

We analysed the Δd_{min} values as a one-way within-subjects ANOVA with 9 levels, and found that there was a significant effect between conditions ($F_{8,152} = 8.828, p = 0.000$). A post-hoc Tukey HSD test found significant differences between `Absent` and `FlashingLights` and `SmallButterfly`, and between `Baseline` and all border types except `SmallButterfly`.

8.4.3.4 Raw Data

Figure 8.19 shows an example of the raw data for a block of trials: a plot of the ***d* values** (vertical axis) over **time** (horizontal axis) are overlaid with annotations indicating when each trial started and ended (i.e. the lines labelled **S** and **E** respectively), the time when participants noticed each highlight (i.e. green lines, labelled **N**), and the name of the border type in use for that trial. The light-blue sections of the plot indicate datapoints that were used for the `Baseline` condition.

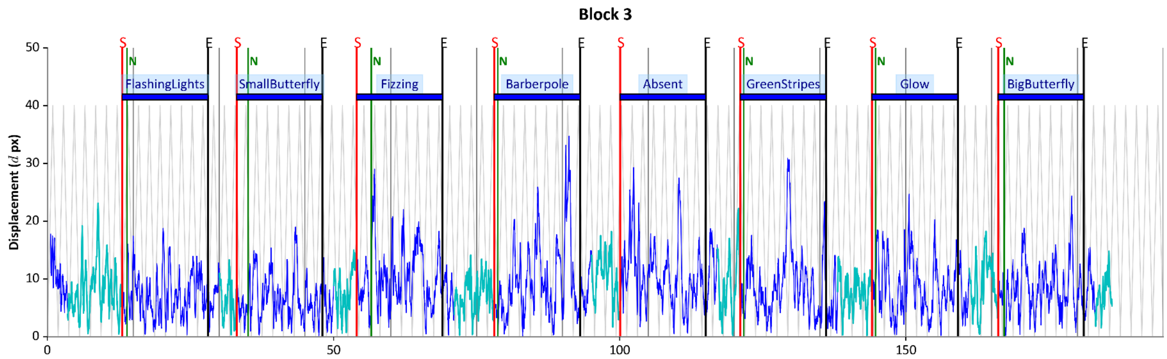


Figure 8.19: *Plot of the raw displacement (**d**) data from one block of trials.*

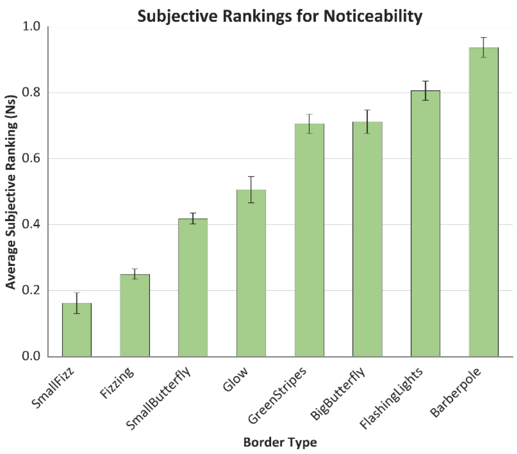
The light-grey lines in the background indicate where the peaks on the cog-shaped path occurred. It can be seen that many of the peaks in the data coincided with peaks in the cog-shaped path.

8.4.4 Subjective Experience

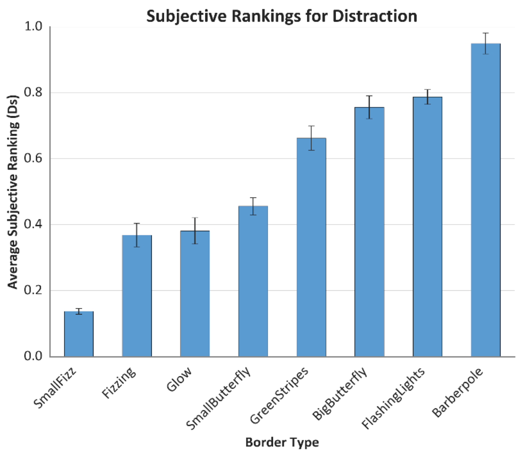
This section presents the results of the Subjective Experience measures. These are drawn from the four-stage post-experiment survey, consisting of two ranking questions, and a series of Likert-scale questions and free-text responses about each border type.

8.4.4.1 Noticeability and Distraction Rankings

Figures 8.20(a) and 8.20(b) show how participants ranked the border types in terms of increasing noticeability and distraction. From these graphs, it can be seen that except for `SmallButterfly` and `Glow`, there was a 1-1 relationship between the ordering of border types in terms of N_s and D_s; this relationship can be seen more clearly in Figure 8.13(b), with a strong correlation ($\rho = 1.000$) between the metrics. Friedman Tests showed that there were significant differences between the Likert-scale ratings across the border types for the N_s rankings ($\chi^2_r = 111.93, df = 7, N = 20, p = 0.000$), and also for the D_s rankings ($\chi^2_r = 108.93, df = 7, N = 20, p = 0.000$).



(a) N_s – Subjective Rankings of Noticeability (normalised). Higher N_s values are more noticeable



(b) D_s – Subjective Rankings of Distraction (normalised). Higher D_s values are more distracting

Figure 8.20: *Plot showing the average subjective rankings for each border type in terms of Noticeability (left) and Distraction (right).*

8.4.4.2 Likert-Scale Survey

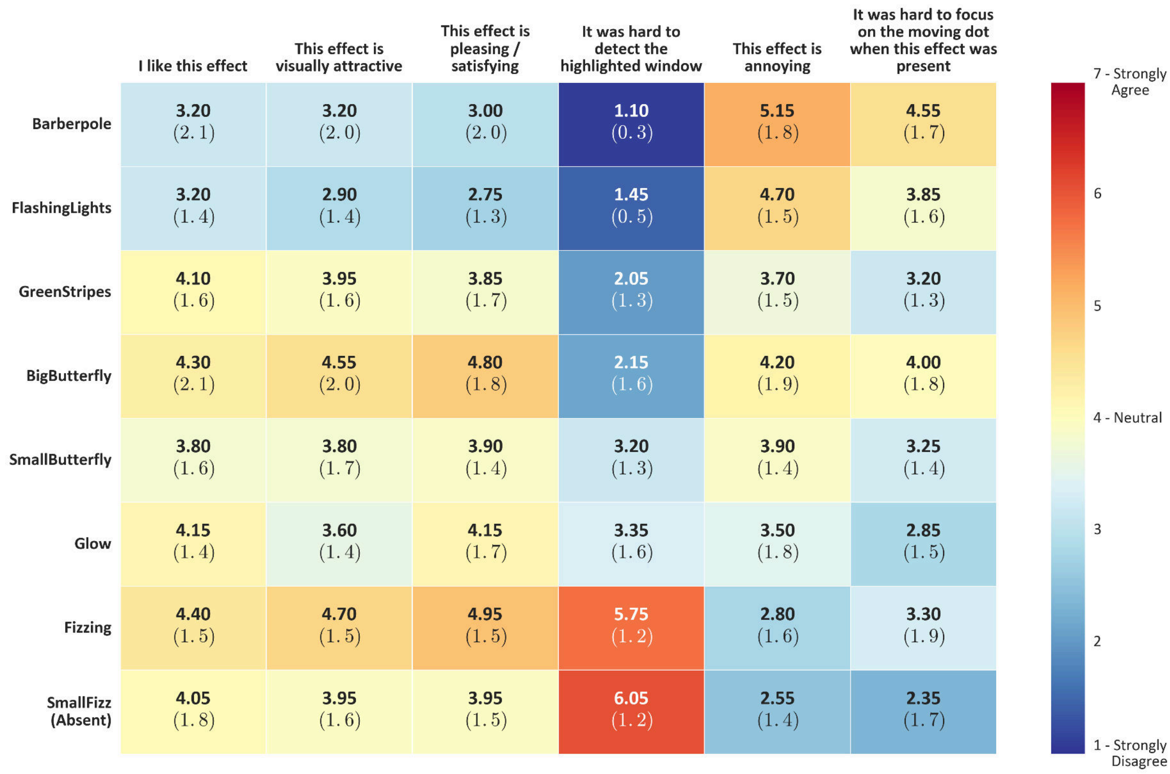
Figure 8.21 shows a heatmap providing an overview of the way that participants answered the Likert-scale questions about each border type. The border types were sorted based on the mean responses to **Question 4** ("It was hard to detect the highlighted window"), as it had the broadest range of responses (from "Strongly Disagree" to "Strongly Agree").

Friedman Tests were used to check whether the responses to each question were statistically significant. Table 8.1 shows that there were significant differences between the border types in all of the questions except the first two (i.e. all except "I like this effect" and "This effect is visually attractive").

Question	χ_r^2	**df**	**p**	
1. I like this effect	10.408	7	0.166590	
2. This effect is visually attractive	17.850	7	0.012665	✓
3. This effect is pleasing / satisfying	27.862	7	**0.000233**	✓
4. It was hard to detect the highlighted windows	91.787	7	**0.000000**	✓
5. This effect is annoying	41.200	7	**0.000001**	✓
6. It was hard to focus on the moving dot …	34.220	7	**0.000016**	✓

Table 8.1: *Results of Friedman Tests for the Likert-Scale survey*

The first three questions were aimed more towards understanding how favourably participants found the effects, while the last three questions were focussed on understanding the negative effects of those techniques. Overall, it can be seen that participant responses across the first three questions were relatively similar, while **Questions 4 and 5** were quite similar to each other. **Question 6** ("It was hard to focus on the dot when the effect was present") was the least similar to all the other measures. The least noticeable techniques (such as



	I like this effect	This effect is visually attractive	This effect is pleasing / satisfying	It was hard to detect the highlighted window	This effect is annoying	It was hard to focus on the moving dot when this effect was present
Barberpole	3.20 (2.1)	3.20 (2.0)	3.00 (2.0)	1.10 (0.3)	5.15 (1.8)	4.55 (1.7)
FlashingLights	3.20 (1.4)	2.90 (1.4)	2.75 (1.3)	1.45 (0.5)	4.70 (1.5)	3.85 (1.6)
GreenStripes	4.10 (1.6)	3.95 (1.6)	3.85 (1.7)	2.05 (1.3)	3.70 (1.5)	3.20 (1.3)
BigButterfly	4.30 (2.1)	4.55 (2.0)	4.80 (1.8)	2.15 (1.6)	4.20 (1.9)	4.00 (1.8)
SmallButterfly	3.80 (1.6)	3.80 (1.7)	3.90 (1.4)	3.20 (1.3)	3.90 (1.4)	3.25 (1.4)
Glow	4.15 (1.4)	3.60 (1.4)	4.15 (1.7)	3.35 (1.6)	3.50 (1.8)	2.85 (1.5)
Fizzing	4.40 (1.5)	4.70 (1.5)	4.95 (1.5)	5.75 (1.2)	2.80 (1.6)	3.30 (1.9)
SmallFizz (Absent)	4.05 (1.8)	3.95 (1.6)	3.95 (1.5)	6.05 (1.2)	2.55 (1.4)	2.35 (1.7)

Figure 8.21: *Heatmap showing an overview of the responses to the Likert scale survey indicating how strongly participants agreed with the statements about each border type. Each cell shows the mean (top line) and standard deviation (bottom line) for the range of responses to each question-border pair. Blue (1.0) represents that participants "Strongly Disagree[d]" with the statement, while Red (7.0) represents that participants "Strongly Agree[d]" with the statement.*

`Fizzing` and `SmallFizz`) were well regarded, with positive ratings for likeability, visual appeal, and pleasantness; they were also considered to be the least annoying and disruptive on task performance. In contrast, the most noticeable techniques (such as `Barberpole` and `FlashingLights`) were not well regarded: participant ratings for likeability, visual appeal, and pleasantness were the lowest among all techniques, while the effects were rated as being the most annoying. However, the responses to **Question 6** ("It was hard to focus on the moving dot") for `Barberpole` and `FlashingLights` were neutral.

Question 5 ("This effect is annoying") was the second most sensitive measure, and appeared to be the inverse of **Question 4**: for instance, the least "hard to detect border" was the most annoying, while the "hard[est] to detect border" was the least annoying. This finding seems like a natural conclusion, since it would be surprising to expect that a difficult to detect highlighting technique was considered to be annoying. However, a clarifying question asked by one of the participants revealed an interesting insight (as well as drawing attention to some ambiguities in the survey). Specifically, there is a difference between "annoyance within the context of the experiment task" and "the nature of the effect makes it annoying". The intention of the survey was to probe the latter meaning. However, an alternative interpretation and direction of inquiry (exposed by the participant's question and subsequent comments) was that participants may have found techniques "annoying" because the techniques made it more difficult to complete the highlight-identification task. For instance, the participant noted that they would have rated `Fizzing` borders as being "the most annoying" because "they were really hard to find".

In **Question 3** ("This effect is pleasing / soothing"), the highest rated techniques (i.e. most pleasing/soothing) were `Fizzing` and `BigButterfly` respectively, while the lowest rated techniques (i.e. the least pleasing/soothing) were `Barberpole` and `FlashingLights`. This appears to support our hypotheses regarding "naturally-occurring" versus "man-made" effects – specifically, that effects that are inspired by "naturally occurring" phenomena are more pleasing and soothing to look at than those that can be considered purely "man-made" effects. Here, both `Fizzing` and `BigButterfly` were based on things found in nature (e.g. fizzing bubbles from carbonated beverages, and butterflies), while `Barberpole` and `FlashingLights` were inspired by borders and signage traditionally used by different various businesses in the real world (e.g. the red-white-blue striped poles spinning outside barber shops, or the large billboards lined with rings of flashing lights outside theatres). However, the `Glow` effect was rated slightly more favourably than `SmallFizz` and `SmallButterflies` (i.e. the low-intensity versions of the aforementioned "nature-inspired" techniques); this is interesting because the `Glow` effect is based more on man-made lighting effects.

8.4.5 Eye Movements – Where was Visual Attention Directed?

Density heatmaps were generated to visualise the distribution of fixation points across the screen. The colour of each bucket/cell in the heatmaps indicated the density of fixation points in that region. This was achieved by plotting the location of each fixation point across all trials and participants, then dividing the resulting 1920×1080 (i.e. Full HD resolution) domain into a grid of hexagonal bins.

8.4.5.1 Overall Fixation Distributions

The heatmap in Figure 8.22 shows the distribution and density of fixations across all trials, for all participants. It shows that participants mostly focussed on following the moving dot, as indicated by the bright circle in the center of the screen. Small clusters of fixations can also be seen around the edges in locations corresponding to candidate windows; these indicate that participants did occasionally look at the candidate windows, instead of only focussing on the primary task.

An interesting feature of the fixation distribution is that participants looked at the area above the circular path (i.e. in the $250 \leq y \leq 400$ region) more often than for any of the candidate windows. Possible explanations for this are:

1. **Calibration Error** – It is possible that errors when calibrating the eyetracker could have resulted in some of the fixations on the upper edge of the path to be shifted upwards. For some participants, the eyetracker calibration on the center-point was vertically offset by 2-3 cm. However, this offset only appeared to be a problem when focussing on the center point directly – in each of these cases, we verified that points within a 5 cm radius of the center-point were being tracked accurately/correctly.

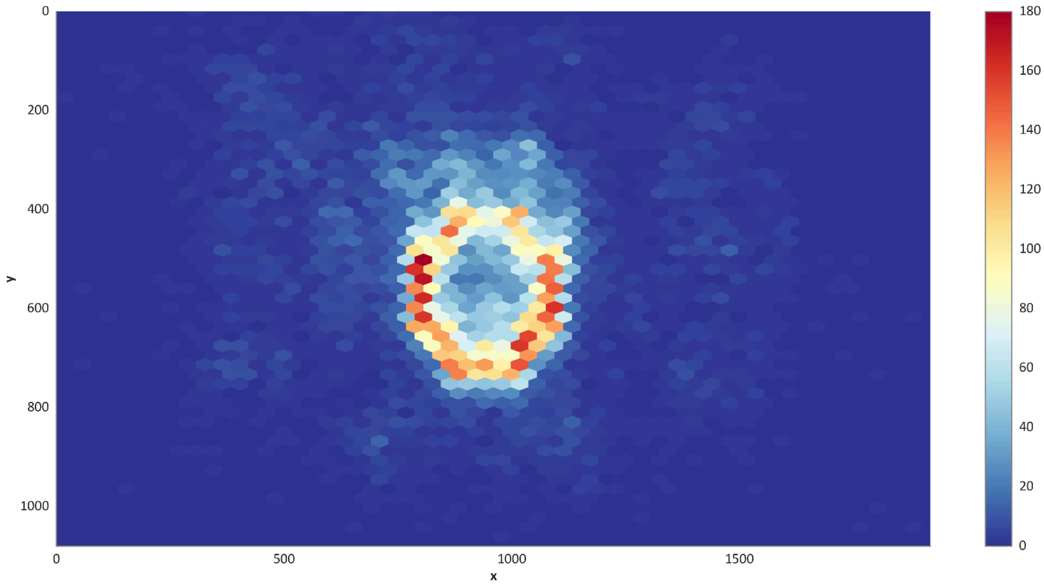


*Figure 8.22: Density heatmap showing the distribution of fixation points across all trials (for all participants). (**Resolution** = 100 bins)*

2. **Increased Motion Sensitivity in Near Peripheral Vision** – Another explanation is that animated border effects may be more distracting/visually attractive when they occur close to the participant's focal point. For example, participants may have been more likely to look at the bottom edges of Windows 2 and 3 if they had the `Barberpole` or `BigButterfly` effects applied to them than if those effects were applied to Windows 5, 8, and 10 instead.

3. **Confounding Effects of the Always-on `Fizzing` Borders** – The `Fizzing` technique applied to the central window may have been a source of distraction (especially along the top edge of the window), due to its close proximity to the focal point, the high colour contrast between the particles and background image along the top edge (i.e. "white-on-dark-blue" near Window 2 versus "white-on-light-green" near Windows 7 and 9), and the visual complexity and/or clutter caused by the titlebars may also have contributed to this phenomenon. A comment from one of the participants supports this interpretation (e.g. "*I usually try to spot it while following the dot as the dot approaches each window.*" – ***p13***)

8.4.5.2 Fixation Distributions Per-Border Type

Figure 8.23 compares the fixation distributions of the border types studied. Overall, it can be seen that the distributions were similar to that shown in Figure 8.22 (i.e. most visual attention is still concentrated on the dot-following task). However, there are notable differences in terms of how many fixations were directed to the candidate windows. Specifically, participants looked at all the candidate windows a lot more in "more distracting" (as per the $||\Delta d_{avg}||$ metric, e.g. `Fizzing` and `Absent (Small Fizz)`) border types than in "more noticeabile" border types (e.g. Barberpole, `FlashingLights`, and `GreenStripes`). These plots help explain the surprising results of the $||\Delta d_{avg}||$ metric as having been caused by participants having to hunt for the borders, instead of having their attention drawn away from their task.

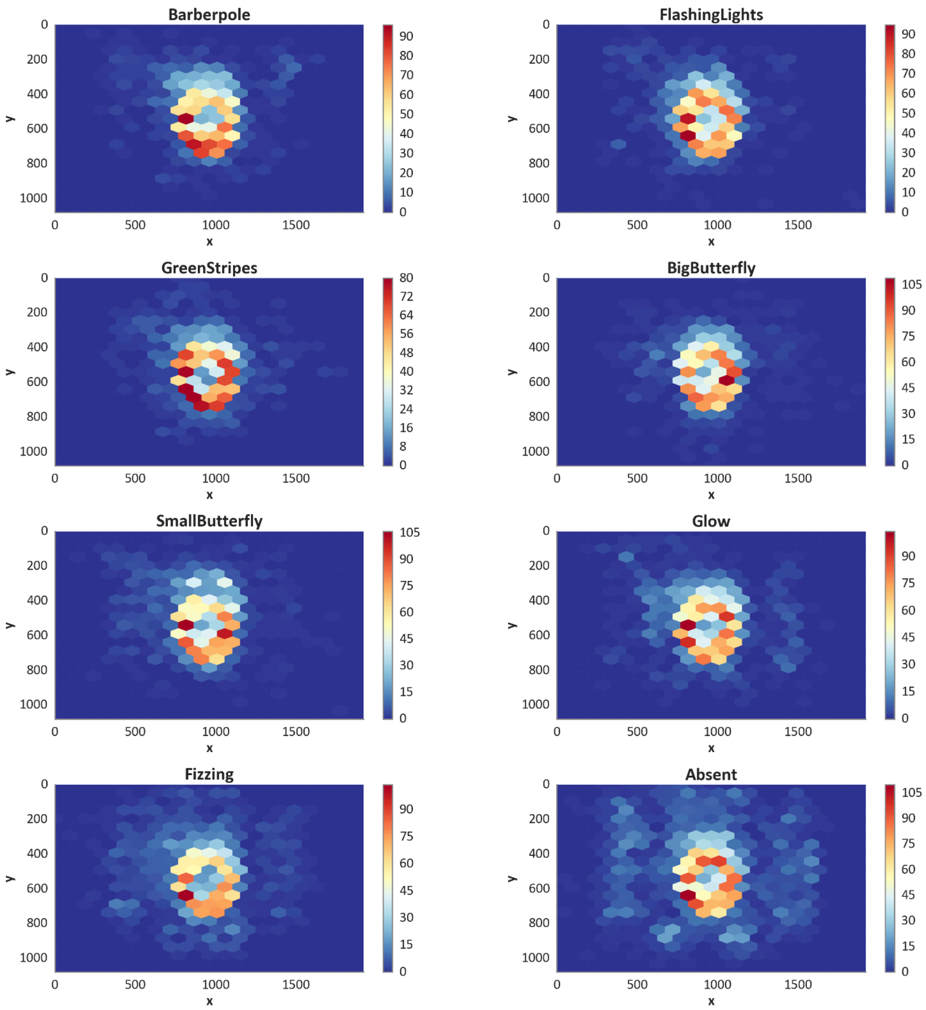


__Figure 8.23:__ Density heatmaps showing the fixation distributions for the different border types. To make the low-density regions more legible, these heatmaps are drawn with a lower resolution of 20 bins.

8.5 Discussion

8.5.1 Summary of Findings

Overall, the results show that the method described can be used to empirically discriminate between different levels of effects corresponding to the "noticeability" and "distraction" of highlighting techniques. Figures 8.15 and Figure 8.18, show that the method could be used to identify several distinct groups of highlighting techniques in terms of each metric. The subjective rankings of noticeability and distraction (N_s and D_s respectively) also showed similar findings (as shown in Figure 8.13(b)).

However, the Noticeability (N_t) and Distraction ($||\Delta d_{avg}||$) metrics were less successful than anticipated.

- **Ceiling Effects** – The Noticeability (N_t) metric suffered from a ceiling effect, as it could not detect significant differences between the three most noticeable techniques (i.e. `Barberpole`, `Flashing Lights`, and `Green Stripes`).
- **Surprising Results** – The results of the Distraction ($||\Delta d_{avg}||$) were the inverse of the expected result (as determined from the Subjective Experience measure for Distraction, D_s). Section 8.5.1.1 discusses these issues in greater depth.

8.5.1.1 Interpretations of the "Distraction" Metrics

The "Displacement" metrics (Δd) were used to measure distraction, by measuring the task performance in the dot-following task. We found that the metrics did not behave as expected, with significant differences between the results of each metric indicating that the metrics were not all measuring the same effect(s) as intended.

The primary metric, **Mean Displacement** (Δd_{avg}), was the most surprising as the results were the opposite of what we had expected. Instead of measuring "Distraction", it appeared to be a measure of "inverse noticeability", with the same relative order of highlighting conditions as **Question 4** ("*It was hard to detect the highlighted window*") from the Likert-scale survey. Furthermore, although differences between the Δd_{avg} values of the conditions could be seen in Figure 8.16, significant differences were only detected between the presence and absence of highlighting, and not between the individual conditions.

The **Peak Displacement** (Δd_{max}) metric was ineffective, as it was insensitive to differences between techniques. No significant differences were detected between most of the techniques as most techniques had similar Δd_{max} values, suggesting that highlighting techniques had little to no effect on the ability of participants to prevent their performance from degrading past a certain point. The task performance feedback (i.e. the size of the dot) may have contributed to limiting or lowering the peak values, since participants would have had strong visual feedback if they strayed further than 40 px from the target, causing them to make abrupt corrections to their behaviour.

Surprisingly, **Minimum Displacement** (Δd_{min}) was a promising metric of distraction. We had expected that there should be no difference between highlighting conditions, as minimum displacement was only expected to occur when no highlights were present. The results show that instead of causing moments of really bad tracking performance (i.e. large spikes in the data) due to momentarily switching attention away from the dot-following task, the "distraction" effect was more like a performance barrier that persistently prevented participants from attaining the same peak performance (i.e. they are not able to get as close to the dot as they could without the distraction).

This suggests that highlighting techniques may instead distract the user by frequently requiring them to make micro-decisions about whether a salient stimulus requires further attention; the cognitive overhead from dealing with this barrage of sensory inputs causes annoyance, fatigue, and delayed responses to other stimuli, as the user increasingly feels that the interface's priorities are misaligned with their own (i.e. the weights of the payoff matrices for the highlight become increasingly negative towards detecting/responding to a signal).

However, caution is needed when interpreting the results of this (Δd_{min}) metric, as although there were significant differences between the conditions (Figure 8.18), the values and differences between them are all in the sub-pixel range (for a task occurring on a screen with thousands of pixels). Also, there are other peculiarities such as `Glow` and `BigButterfly` featuring as techniques with the highest Δd_{min} values (despite these not being the highest rated techniques under any other metrics). Further work is needed to determine if these findings are reproducible.

8.5.1.2 Source of Distraction

Questions 4, **5**, and **6** of the Likert-scale survey were designed to investigate potential reasons for why highlighting techniques were perceived to be distracting. Figure 8.21 shows that the responses to **Questions 4 and 5** were generally the inverse of each other – that is, an easy to notice technique (e.g. `Barberpole` or `FlashingLights`) was likely to be very annoying, while a hard to notice technique (e.g. `Fizzing` and `Glow`) was unlikely to be annoying. These results make sense as it would be unusual for a barely noticeable highlight to be able to distract the user.

The relative ordering of highlighting techniques in **Question 4** resembles the ordering from the Mean Displacement (Δd_{avg}) metric. Although the results appeared to show more of an "inverse noticeability" effect, it could still be considered to be a measure of distraction in that it satisfied the criteria outlined in Definition 4: That is, the Δd_{avg} metric measured varying levels of *performance degradation* when participants were exposed to different highlighting techniques. Thus, we argue that an "inverse noticeability" effect (i.e. task performance is degraded when the user is looking for the presence of a highlighted item, but have difficulty doing so as the highlight is difficult to detect/notice) can be considered as a type of "distraction".

The responses to **Question 6** ("*It was difficult to focus on the dot-following task*") did not appear to be closely related to any of the other metrics (performance-based or subjective). From Figure 8.21, it can be seen that although there was some similarity between the relative ordering of the techniques in **Questions 4** and **6** (i.e. `Barberpole` and `SmallFizz` were the first and last techniques in both rankings) there were a few outliers (e.g. notably `BigButterfly`, `Fizzing`) where the techniques were rated as making it harder to focus on the primary task. A possible explanation can be seen from the responses to **Question 3** ("*This effect is pleasing/satisfying*"), where both `BigButterfly` and `Fizzing` scored highly, relative to the surrounding techniques. This suggests that participants may have found it hard to focus on the primary task when these techniques were shown, as their "pleasing/satisfying" nature may have made participants want to look at them. Comments made about the `BigButterfly` appear to support this hypothesis. For example, a participant commented that it was "*[A] very attractive border.* ***At times [I] was distracted from the task of following dot to look at it***" (***p13***). It is possible that this could have occurred "*because [the butterflies] are large and flappy once they have faded in I find it hard to ignore them*" (***p19***).

8.5.1.3 Reliability of Subjective Experience Measures

Compared with the performance-based measures of noticeability and distraction, the subjective experience measures in this study appear to be more reliable metrics of noticeability and distraction. For example, the subjective measures of noticeability (N_s and Question 4 from the survey) did not suffer from ceiling effects (which reduced the utility of the N_t metric).

As in the previous study (Chapter 7), there was still some ambiguity in the wording of some of the questions. For instance, we realised that Question 1 of the survey (i.e. "*I like this effect*") can be interpreted in two ways: 1) In terms of the general likeability of the technique (i.e. the intended interpretation), and 2) In terms of the how conductive the technique was for tasks in the experiment. The difference between these interpretations can be understood from some of the participant comments about the `Barberpole` and `Fizzing` border techniques:

- `Barberpole` – Participant comments indicated that they did not really like the look of the effect and would not want to use it, while indicating that they really "liked" it in the context of the experiment as it made it easy to perform the task (e.g. "*You cannot miss it. It's big, bold, and ugly.*" (***p9***), "*This looks awful. I hate it. But it's definitely effective at getting my attention.*" (***p19***), "*Very attractive for some odd reason. Easy to spot.*" (***p13***)).
- `Fizzing` – In contrast, participants appeared to "generally like" the effect itself (e.g. "*Nice and gentle. Gets out of the way.*" (***p2***), "*This looks pretty cool! I like it. ...*" (***p19***)). However, many commented that within the context of the experiment, they really disliked it because it made it difficult to find the highlighted window (e.g. "*Difficult to spot amongst the other "fizzing" borders.*" (***p13***))

Another issue with the subjective experience measures was that the highlighting techniques were presented in different ways (e.g. compare Figure 8.9(b) (during trials) and Figure 8.11 (survey layout)). During the trials, highlighting techniques were presented in relative isolation, whereas during the survey, all techniques were shown on screen at once and in close proximity to each other. Furthermore, performance-based metrics were collected during/immediately after each exposure, while subjective experience metrics were only collected at the end of the whole experiment (after multiple exposures). These limitations should be noted when comparing the two sets of metrics, as one participant commented on how different the techniques felt in the two scenarios, while another commented that their feelings about the techniques changed over the course of the experiment (e.g. while they initially liked some effects, after a few exposures, they found these annoying or frustrating to work with).

8.5.1.4 Short-Term versus Long-Term Suitability for Deployment

Comments made by participants revealed how their perceptions of some techniques changed over the course of the experiment. For example, participants initially found `Barberpole`, `BigButterfly`, and `FlashingLights` quite "novel", "nice", and "fun" to use. However, they would begin to tire of those effects as the experiment proceeded (e.g. "*Noticeable and kind of nice at first, quickly becomes annoying*" (***p4***), and "*Pleasing when looking [it] at [for] the first time, but not sure when looking at it so many times*" (***p7***)).

This raises the issue of how frequently and for how long highlighting techniques should be deployed. For instance, perhaps the use of annoying but novel effects (e.g. `Barberpole` or `BigButterfly`) should be restricted to interfaces that the user may only encounter periodically. For example, a `BigButterfly` or `FlashingLights` effect could be used in the interface for a "walk-up and use" kiosk to help guide the user through an unfamiliar process while providing a "novel" and "fun" experience. In contrast, less distracting techniques (e.g. `Glow`) could be useful for interfaces that frequently used.

8.5.1.5 `Barberpole` versus `GreenStripes` – The Effect of Colour Contrast

The `Barberpole` and `GreenStripes` techniques both featured thick borders with multi-coloured stripes travelling in a circular motion around the border at a moderate speed, with the only difference between the two techniques being the colour schemes used. `Barberpole` featured a high-contrast "red-white-blue" colour scheme (as traditionally seen on signage outside barbershops, or on envelopes), whereas `GreenStripes` featured a low-contrast pair of light-dark "teal/blue-green" shades. While the performance-based metrics were unable to discriminate between the two, Figure 8.21 shows that overall, participants favoured `GreenStripes` over `Barberpole`.

Many participants expressed concern about the high-contrast colour scheme used by `Barberpole`, commenting that: "*It's big, bold, and ugly.*" (***p9***), "*This looks awful. I hate it.*" (***p19***), and "*Very annoying, and too bright. Would be horrible at night time in a dark room.*" (***p18***).

In contrast, most responses about `GreenStripes` expressed neutral-to-positive sentiments about it (e.g. "*I like this best*" (***p12***), and "*Soothing, I quite like this one. Stood out well*" (***p9***)).

8.5.1.6 `BigButterfly` versus `SmallButterfly`

The `BigButterfly` and `SmallButterfly` techniques both used a particle system to display animated sprites depicting butterflies flapping their wings. The `BigButterfly` effect used larger sprites (60px vs 25px), with a more energetic flapping motion ($\pm$50% vs $\pm$10% vertical movement). Most comments noted how `SmallButterfly` was less pleasant and less effective than `BigButterfly` (e.g. `SmallButterfly` was: "*Not very attractive, and also not overly noticeable*" (***p9***), and "*Too small, not as pleasant as large butterflies*" (***p10***)).

In contrast, comments about `BigButterfly` were polarised: participants either really liked the technique (e.g. ***p2***, ***p7***, ***p10***, ***p13***, and ***p19***), or they found them "*quite distracting*" (***p3***), "*very annoying*" (***p4***), or "*gross*" (***p9***).

Participants who liked the `BigButterfly` effect cited two main reasons:

1. **Visual Appeal** – Several participants noted that the effect was quite visually pleasing or appealing. Comments included: "*These are visually appealing, and oddly satisfying*" (***p19***), "*Pleasant to look at*" (***p10***), "*It's really soothing and gentle*" (***p2***), and "*Artistic effect. Pleasing when looking at the first time, but not sure when looking at it so many times*" (***p7***).

2. **Gentle Fade In** – One of the aspects most liked was how the butterflies appeared to fade in. For example, "*I like the way they fade in. It means I'm not immediately pulled away from what I'm doing.*" (***p19***)

The flapping motions may have been perceived by some participants to be quite distracting (e.g. `BigButterfly` is "*[a] very attractive border. At times [I] was distracted from the task of following dot to look at it*" (***p13***). This could have been "*because they're large and flappy once they have faded in I find it hard to ignore them*" (***p19***).)

8.5.2 Limitations, Sources of Error, and Review of Design Decisions

In this section, we discuss a number of experiment design decisions which may have affected the sensitivity and validity of this experiment.

8.5.2.1 Rationale for Design Choices in the Primary Task Design

The dot-following primary task used in this study was chosen to avoid the unpredictability of prior methods used in the literature. We specifically wanted to reduce the non-deterministic (i.e. unpredictability) aspect arising from having the target performing random walks. This was necessary to improve the sensitivity of the method, as pilot testing of the traditional "random-walk" techniques identified significant amounts of noise (Appendix A.2.1).

There was a fine balance between ensuring the participants remain engaged, and the task being too difficult for humans to accurately perform. Here are some of the key concerns/challenges we had to overcome, and reasons for the design of the final task:

- **The tradeoff between number of peaks, total time, and movement speed** – The final configuration of the task featured 16 points (or 8 peaks) over 15 seconds. This configuration was a compromise between two extremes. For instance, if the task was too *easy* (e.g. 12 points / 6 peaks over 20 seconds), participants would often move faster than the target. However, if the task was too *difficult* or physically demanding (e.g. 24 or 50 points over 20 seconds), participants would struggle to keep up, and would tire quickly from the intense concentration and hand-eye coordination required to complete the task.
- **The path should be visible** – We briefly experimented with hiding the path in the belief that having the path visible may have been making the task too easy. However, with the path hidden, the task became prohibitively difficult to perform (even knowing what the path looked like beforehand).
- **The need for feedback/gamification** – It was easier for participants to stay on task when they have some feedback indicating how well they are performing the task. By introducing a feedback mechanism (i.e. the size and colour of the moving dot), we found that the task was more engaging as it introduced a gamification aspect to the task.

- **Clutching, Path Radius, and Starting Position** – The current r_{min} and r_{max} values were calibrated to allow most participants to be able to complete the entire task without having to pause and change hand position. Furthermore, we found that it helped to make participants start moving from the top of the path (i.e. when their arm at the maximum extention point for depth) to avoid clutching issues.

Although we settled upon a suitable configuration for this experiment, there are opportunities for subtle variations in the design of the task which may help reduce fatigue and help participants remain more engaged. For example, instead of using the same shape in each block, using a slightly different shape in each block (e.g. by varying the number of peaks, varying the radii, or simply rotating the shape) may have helped participants stay engaged. This is important because there was some evidence that participants may have been getting fatigued – although not evident in the final results (where dot-following performance in blocks 1-4 was relatively constant), there was a notable U-shaped curve (i.e. dot-following performance peaked in block 2, before dropping off) in the data for the first 11 participants.

8.5.2.2 Constant Presence of Fizzing Border Effects

All windows had a "Fizzing Border" effect applied to them throughout the experiment. This was done to investigate whether it is possible to have a desktop environment where there is a constant baseline of non-essential animation present. Two levels of the fizzing borders effect were used: the stronger `Fizzing` technique was used on the central primary-task window, while the weaker `SmallFizz` effect was used on all peripheral windows (except for the window that was highlighted during a trial).

For most of the highlighting and baseline conditions, the constant presence fizzing effects appeared to have little effect on the ability of participants to successfully perform the dot following primary task or to notice the appearance of highlights on a candidate window. It is not possible to determine whether performance (relative to conventional "static" interfaces) was affected by the constant fizzing effects, due to a lack of control conditions for those. However, this experiment showed that even with constant fizzing effects applied, participants could still perform the experiment tasks with considerable levels of performance.

The most notable negative effects of the constant fizzing border effects were in the `Fizzing` and `Absent (SmallFizz)` conditions. These interactions could be attributed to several factors:

1. **Same Technique** – All the windows already had an fizzing border effect applied to them. The primary-task window had the `Fizzing` technique (i.e. the "active" version) applied, while the 9/10 candidate windows were showing the `SmallFizz` technique (i.e. the "inactive" version). It is possible that because all windows had this effect applied already, participants were more inclined to ignore any "similar looking" effect. This could happen if their visual-attention system considered the difference between the "active" and "inactive" versions of the effects too similar for there to have been any change in stimuli (i.e. relative to the prior stimuli, the change was considered to be a random glitch [65]), and thus the mental models for that part of the scene would not need updating yet [139]. Notably, one of the participants (***p21***) uttered, "[it's] still going", in relation to a `Fizzing` border that had disappeared two seconds earlier.

2. **Gradual Appearance** – Unlike many of the other effects, the `Fizzing` effect did not occupy a solid region of screen space that was immediately filled with a solid colour (or colours) as soon as the window became highlighted. Instead, as the effect was formed using a particle system, it took 2-3 seconds for the newly emitted particles to accumulate enough to reflect the "active" state. As a result, there may have been less of an initial pop-out effect to catch the user's attention. Participant comments about some of the other techniques with gradual onset suggest that this may have been a factor (e.g. some participants noted that with `BigButterfly` and `Glow`, they were unsure whether they were supposed to only respond to these effects when they had "fully appeared" as opposed to when they were "still fading in"). This effect could be viewed as a type of "inverse Ephemeral Adaptation" [64]: instead of drawing attention to the highlighted item by making it appear earlier than any of the other non-highlighted items, the gradual appearance of the "highlight" serves to make the affected item less noticeable than it otherwise would be.

3. **Nature of the Effect** – It is of course possible that the nature of the effect (i.e. a stream of particles slowly being emitted from an object and fading out) is a generally subtle effect that does not draw much attention to itself.

8.5.2.3 Highlights Present Indicator

Participants were given a visual prompt (i.e. an indicator in the middle of the primary task window, as shown in Figure 8.9(b)) when each trial started and ended (i.e. when a window may have been highlighted). As a result, care is needed when interpreting the noticeability results: it is important to remember that the results apply only when the user has been made aware of that a window has been highlighted (e.g. using an visual indicator like in this experiment, or by a sound effect/chime typically associated with the appearance of dialog boxes [103]).

This indicator was used to preserve the integrity of the experiment. During early pilot testing of the experiment protocol (see Appendix A), we found that it was often very difficult (and in some cases, impossible) to notice that a window had become highlighted using the `Fizzing` technique, especially if the window in question was located far from the center of the screen (where the participant's focal point was assumed to lie). This was problematic because if participants completely missed a hard-to-detect technique, there would be insufficient data for those conditions (given the within-subjects study design). Furthermore, we found that in the absence of any indications of when trials started/ended, participants would start hunting for highlighted windows, even if a trial was not currently running, thus affecting the *baseline performance* measures. By using a visual indicator, these problems were averted, while also increasing the ecological validity of the study (given that "alerts/notifications" in the real world are often deployed in conjunction with secondary cues such as beeps, chimes, and/or haptic feedback).

Alternatively, we could have just made each trial longer (i.e. setting a weaker Noticeability Cutoff Threshold (Section 6.4.2) to give participants a greater chance of noticing the less noticeable techniques). However, that would have also greatly lengthened the overall experiment duration, resulting in greater fatigue (from having to spend longer periods of time concentrating on the primary task), which would have decreased the validity of the distraction measurements.

However, the usage of the indicator did present some issues about the external validity of some noticeability results. Notably, the indicator made hard-to-notice techniques (e.g. `Fizzing`) appear to be more noticeable than they would naturally be (i.e. since participants would try harder to detect a highlighted window), while also being the likely cause of the high levels of performance degradation in the `Absent` condition (as participants would start looking for a highlighted window that was not present, they would waste time and effort, causing the dot-following accuracy to drop while "distracted"). Several participants (e.g. ***p19*** and ***p21***) indicated that they only managed to find the `Fizzing` borders in some cases because the prompt alerted them to its presence.

8.5.2.4 Confounding Impact of Mixing Noticeability and Distraction Tasks

It is possible that some component of the "distraction" effect found may have been caused by participants performing the secondary task (i.e. responding to the highlights), instead of this having been caused by the highlighting effect. This is because to respond to a highlight, the following sequence of things had to happen:

1. Participant notices or becomes aware of the highlight
2. Participant remembers/realises that they they need to carry out an action to respond to this event
3. Participant initiates and performs button-pressing action
4. (Participant "recovers" from this upset, and refocusses on the primary task)

In the first block or two, participants were still learning to perform the task, and would likely have had to consciously remind themselves of what they needed to do (Step 2). This could have caused some attention to be directed away from following the dot, as participants deliberated or hestitated on what to do, and may also have slightly inflated noticeability times in those blocks. For this reason, data from the first block of trials was not included in the data analysis, and (participants were told that "the first block is just a practice run".

Another issue affecting this experiment and other dual-task studies is that multi-tasking skill/ability varies between participants. Participants who have less multi-tasking/hand-eye coordination and motor skill may have (unconsciously) stopped following the dot when focussing on performing their other task. This would in turn have allowed the dot to get further away from the cursor, resulting in a need to catch up. Is is also possible that the task switching involved may have incurred a small amount of "distraction" overhead to the participant, requiring a second or so to recover from (e.g. realising that the dot is had moved further than expected, overcompensating, and then finally getting back in sync).

In defence of this experiment design, by combining the measurement of noticeability and distraction measures in the same trial, we could have a shorter experiment (i.e. there are only half the number of trials required), since there would not need to be blocks of noticeability trials followed by distraction trials. While that would have avoided this problem, there may still have been a related problem (i.e. in Step 2 of the reaction process, there is a greater cognitive load for participants to remember whether they should be responding to highlights or completely ignoring them), which may have had other undesired side effects. Furthermore, we realised that forcing participants to complete "pure distraction" trials would not have yielded much more meaningful insights about the effects of these highlights than considering the later-parts of each trial (e.g. the last 5-10 seconds of each trial), since in theory, once a highlight has been noticed and appropriately handled, participants should have been ignoring/trying to ignore the highlights.

8.5.2.5 Ceiling Effects

The noticeability metric (Time to First Noticed, t_n) suffered from a ceiling effect, whereby it was not possible to distinguish between the three most noticeable techniques (`Barberpole`, `FlashingLights`, `GreenStripes`). This was likely the result of these techniques being so noticeable that the limiting factor became the speed at which participants could respond using their hands, as opposed to there not being any difference between these.

This raises the question of whether it would be possible to avoid such issues. One approach (inspired by our previous experiment), would be to artificially make the task more difficult to perform, in an attempt to stress the visual processing system so that we may be able to distinguish between techniques at the higher end with greater sensitivity. Examples of things that could be tried include having more candidate windows in total, changing the sizes of the windows, changing the distances of the windows from the center of the screen (and/or changing the size/resolution of the displays), or introducing more visual disturbances (e.g. in the form of salient window contents). However, applying these manipulations comes with the risk that it decreases the external validity (in particular, realism) of the experiment.

Another possible solution would be to investigate using other metrics. In the case of the noticeability metrics, it appears that the Δd_{avg} metric acts as second measure of noticeability, and is one that appears to be slightly more sensitive than t_n was.

8.6 Conclusions

This chapter presented a dual-task experiment for measuring the noticeability and distraction characteristics of 8 highlighting techniques (in the form of animated window borders, or AWB's). Participants used the mouse to track a dot moving around a cog-shaped path, while responding to the presence of highlighting techniques by tapping the spacebar.

The results showed that it was possible to use this method to measure the noticeability and distraction characteristics of AWB's. Some of the metrics (e.g. *Time to First Noticed* (N_t), and *Minimum Displacement* (Δd_{min})) were sensitive enough to detect significant differences between conditions at multiple levels of performance. However, other metrics (e.g. *Mean Displacement* (Δd_{avg}) and by extension, the *Distraction* metric ($||\Delta d_{avg}||$)) were less effective than expected.

Part III

Discussion and Conclusion

9 Discussion and Directions for Future Work

Highlighting techniques are a diverse class of visual communication techniques that make users aware of salient information in a timely manner. They are commonly used across a wide range of user interfaces. However, there has traditionally been a lack of understanding about how to select, apply, and control their effects for the best results. To reduce this knowledge gap, this thesis has developed a structured framework for describing highlighting techniques, and experiment methods to empirically analyse and compare the relative quality of highlighting techniques.

Specifically, the work presented in this thesis provides the following contributions:

- A review and summary of foundational insights (Chapter 2) and prior studies (Chapter 3) of highlighting techniques/effects across the HCI and Psychology literature;
- The development of the PCCH framework for describing/modelling highlighting techniques in terms of how they are constructed and controlled (Chapter 4);
- The development a new empirical framework for measuring and comparing the relative quality of highlighting techniques in terms of their emergent Noticeability and Distraction characteristics (Chapter 6);
- The development of two experiment protocols and running user studies to verify those methods. In the process, we also gathered performance-based and subjective experience data about representative examples of commonly used highlighting techniques (Chapters 7, 8).

However, it should be noted that the work presented in this thesis represents a step towards increasing our knowledge of highlighting techniques. There are many opportunities for future research to use, extend, and/or further develop the methods presented in this thesis. These range from using the experiment protocols presented to characterise techniques across the design space identified by our framework, to developing new experiment methods to investigate promising directions of inquiry identified/exposed by our work.

This chapter takes a higher level look at the progress the work presented in this thesis has made towards bridging the knowledge gap, examines some issues exposed by this work, and provides suggestions of promising directions for future research.

9.1 Progress Towards Objectives

As outlined at the start of this thesis in Section 1.1, we identified the following key opportunities for research contributions to bridge the lack of understanding about highlighting techniques and their effects:

1. There needs to be a **structured framework** for *describing* the **control and construction**

of highlighting techniques.

2. There needs to be an **empirical method** for *measuring* both **noticeability and distraction** of highlighting techniques.
3. There needs to be more reusable **empirical data** about the noticeability and distraction effects of highlighting techniques, so that designers can use/refer to this for objective design guidance.

Chapter 1 also outlined criteria for judging successful completion of these objectives. The progress made by the research outputs towards each objective are now discussed.

9.1.1 Progress towards Objective 1 – Design Framework

OBJECTIVE 1 was to develop a framework/system that can be used to accurately describe the wide range of highlighting techniques. As stated in Sections 1.1 and Section 4.2, to successfully complete this objective, the resulting framework should be able to objectively and unambiguously describe visual effects for both human and computer comprehension, provide insight into how the effects may interact with each other (i.e. specifically, how they may be able to be combined with other effects), and be pragmatic (in terms of issues related to the practical implementation of these effects).

PCCH (our structured framework for Interactive Highlighting Techniques, presented in Chapter 4) satisfies these criteria. It was inspired by modern UI toolkits such as QtQuick/QML and the animation systems used many in content creation tools (e.g. Blender, Maya, and After Effects). As discussed in Section 4.2, highlighting techniques can be described in terms of how their effects are constructed and controlled: that is how they are constructed from layers of Visual Elements or "Pixel Buffers" (Section 4.3), the parameters (or variables/settings) of the Visual Elements and the Object-Level/Pixel-Level manipulations applied to those elements, and how the values of those parameters are controlled over time (Section 4.4) to achieve highlighting effects. This framework satisfies the stated criteria by providing a standardised model and language for describing highlighting techniques, based on concepts commonly found in the UI toolkits used to implement them (but without being directly tied to any particular technology/toolkit). Section 8.2 shows how the ideas in our framework can be used and expanded upon to characterise the design space for a new class of highlighting techniques (i.e. for Animated Window Borders).

9.1.2 Progress towards Objective 2 – Empirical Methods for Measuring Noticeability and Distraction

OBJECTIVE 2 was to develop an empirical method for measuring the noticeability and distraction effects of highlighting techniques. The work presented in Chapter 6 discussed high-level principles for what such methods require, such as the need to measure both Noticeability and Distraction simultaneously using performance-based measures, that the objective measures needed to be sensitive and reliable, that the method is pragmatic to use and that it is externally valid.

Using these principles, we developed and conducted 2 user studies (Chapters 7 and 8) to analyse the noticeability and distraction of common highlighting techniques. Chapter 7 used a path-tracing method to characterise commonly used highlighting techniques in an abstract visual search task. Chapter 8 used a dual-task method to characterise animated window border effects. The noticeability and distraction metrics in both studies could be used to successfully detect significant differences between the highlighting techniques for both Noticeability (N_T and N_t) and Distraction (D_T and $||\Delta d||$) metrics. However, the method from the second study (Chapter 8) appeared to be less successful, as although there were significant effects, the metrics did not work as well as expected (e.g. N_t was insufficiently sensitive as it suffered from ceiling effects, while $||\Delta d||$ appeared to measure "Inverse Noticeability" instead of "Distraction" as intended).

9.1.3 Progress towards Objective 3 – Data on Highlighting Effects

OBJECTIVE 3 was to gather empirical data on the noticeability and distraction characteristics of highlighting techniques, so that designers could refer to this data for design guidance. The data from the two studies presented in this thesis represent only the first steps towards understanding the broad design space for highlighting techniques (as outlined in Chapter 4):

- Study 1 (Chapter 7) analysed 4 types of highlighting techniques: 3 were commonly used highlighting techniques (i.e. `Colour`, `Pulse`, `Shake`) and one was an experiment technique (i.e. `ShootingStar`). These techniques were analysed at two levels of intensity: *Low* (i.e. low contrast, low amplitude/size of movement, and slow movements (e.g. at 2 Hz repeat frequencies), and *High* (i.e. high contrast, large amplitude/size of movement, and fast movements (e.g. at 6 Hz repeat frequencies)). Each *Highlight Type* $\times$ *Strength* combination was analysed with 16 and 64 item grids, and at 2 different distances from the initial focal point.
- Study 2 (Chapter 8) analysed 8 types of Animated Window Border effects, to evaluate how well they could be used to catch the user's attention when the user was engaged in another task. The effects studied included 4 particle-based effects (`Fizzing`, `SmallFizz`, `BigButterfly`, and `SmallButterfly`), 2 moving striped patterns (`Barberpole` and `GreenStripes`), an effect inspired by traditional borders on signage (`FlashingLights`), and a commonly used highlighting technique on borders (`Glow`).

While these studies functioned more as "proof of concept" tests of the empirical methods developed, they still provide valuable insights for designers about the highlighting technique landscape, as some data is better than no data. Study 1 (Chapter 7) sampled two points along each of the main parameters for each technique, while Study 2 (Chapter 8) performed point samples within the larger design space.

However, there is still a great need for further studies to refine the understanding of the design space that these two studies have identified, by densely sampling the "in-between" points along each axis, to provide a clearer picture of the effects of manipulating each parameter. Preliminary tests indicate that there are non-linear relationships/interactions between these parameters. This suggests that simply encoding these findings in the form of design guidelines may not adequately account for the complex and subtle relationship between parameters.

There are also opportunities to refine the methods presented and/or to develop other methods for measuring the noticeability and distraction of highlighting techniques. More details about these are presented in Section 9.4.

9.1.4 Outcomes of Key Hypotheses

In relation to Objectives 2 and 3, this thesis investigated the following main hypotheses (as outlined in Section 1.2.3) with the following results:

1. **Noticeability and Distraction Can Be Objectively Measured** – ✓*Confirmed*

   Studies 1 and 2 (Chapters 7 and 8) found evidence supporting **H1.1**. They showed that performance-based measures of Noticeability and Distraction can be used to identify significant differences between highlighting techniques at multiple levels of Noticeability and Distraction. This result means that the methods presented in this thesis successfully satisfy Objective 2.

2. **More Noticeable but Less Distracting Techniques Exist** – ✓*Confirmed*

   A secondary purpose of Studies 1 and 2 was to show that it is possible to find a pair of highlighting techniques, H_x and H_y, such that H_x is more noticeable and less distracting (i.e. superior) to H_y. Both studies showed that this is possible:

   - In Study 1 (see Section 7.3.1.1), both the performance-based and subjective experience metrics for Noticeability and Distraction successfully identified multiple pairs of highlighting techniques where this property holds. For example, the performance-based metrics (N_T and D_T), showed that `HShootingStar`, `LShake`, `HPulse` were all more noticeable and less distracting than `LPulse`. The subjective-experience measures (N_S and D_S) showed that participants considered `HColor` to be more noticeable and less distracting than `HPulse`, `HShake`, and `HShootingStar`.
   - In Study 2 (see Section 8.4.1), the plot of subjective-experience measures (i.e. N_s versus D_s, as shown in Figure 8.13(b)), `Glow` was more noticeable and less distracting than `SmallButterfly`. However, no such conclusions could be drawn from the performance-based measures due to the problems with the $||\Delta d_{avg}||$ distraction measure.

3. **Objective and Subjective Measures of Noticeability and Distraction Have the Same Results** – $\times$ *Mixed/Negative Result*

   As a sanity check for the objective (performance-based) metrics, both studies also investigated the hypothesis that the objective and subjective metrics should produce similar results. However, we did not find clear evidence for this being the case:

   - In Study 1 (see Figure 7.9), there were few similarities between the objective and subjective metrics. According to the objective metrics (N_T), `HPulse` was the most noticeable, while the subjective metrics (N_S) found that `HColor` was the most noticeable instead. Similarly, the most distracting techniques were `LPulse` (for the objective metric, D_T) and `HPulse` (subjective metric, D_S).

- In Study 2 (see Figure 8.13), the two sets of metrics were the complete inverse of each other. After inverting the $||\Delta d_{avg}||$ axis (as in Figure 8.14, there were greater similarities between the two sets of metrics. However, it is harder to physically justify such a manipulation. Therefore, this result is only weak evidence in favour of this hypothesis.

9.2 Research Generalisability

This section discusses what insights the HCI community can gain from the work presented in this thesis, and of possible limits to the generalisability of these findings.

9.2.1 Significance and Relative Merits of Our Design Framework

Our PCCH design framework (Chapter 4) provides a considerable improvement over prior frameworks in terms of multiple aspects:

- **Scope/Oversight** – Our framework can be used to describe all visual effects or manipulations applied to a widget (or similar visual components forming the graphical user interface of a piece of software, including buttons, menus, windows, and even the cursor). It also describes how these effects can be integrated together. Our framework achieves this by providing better coverage of the highlighting technique design space than the prior frameworks identified in Section 4.1 by being a superset of all of the concepts identified in those prior frameworks. Also, Section 8.2 demonstrates how the general principles of our framework can be extended/specialised for particular classes of highlighting techniques (i.e. for Animated Window Borders in Chapter 8).
- **Practicality** – Our framework is directly inspired and informed by constructs commonly found in current state of the art UI toolkits and content creation packages (such as Blender, Maya, QML/QtQuick, HTML5/CSS3, Swift/Quartz, JavaFX, and After Effects). This means that there is lower translation-gap required between human-understandable descriptions of the techniques, and the mechanics of their implemention. This has potential benefits for optimising the iterative design-test-review workflows of designers, by shortening the *implementation* step when testing design ideas.
- **Precision of Description** – Our framework provides a vocabulary of constructs for creating/constructing common types of highlighting effects. It also introduces the concept of HL Parameters (i.e. settings/inputs for different parts of the components of a highlighting effect) as a formalised system for describing how a highlighted widget is manipulated to create and control the highlighting effects applied to it. This is in contrast to many traditional frameworks (e.g. [102, 37, 106, 153]), where levels of highlighting intensity are often described in vague and coarse-grained terms (e.g. “low”, “medium”, “high”).

These properties make our design framework well suited to being used for characterising highlighting techniques, organising/managing records of their empirically-determined effects, and for helping designers/researchers identify classes of techniques not yet explored.

The use of HL Parameters to both describe and control the effects of highlighting techniques is instrumental for these purposes. The combination of parameters used determines the *multi-dimensional space of effects* that can be achieved using a particular technique. For example, the `HPulse` technique used in Chapter 7 has two parameters (*Frequency* and *Amplitude*); these design space formed by these parameters can be visualised as a two-dimensional grid/surface, where the intersecting gridlines represent different combinations of parameters, and the vertical/height (Z-axis) of each point represents the effectiveness (i.e. Noticeability/Distraction) effects of this combination. Furthermore, these multi-dimensional spaces are necessarily *bounded*, as there are physical limits on the range of each parameter (i.e. *Frequency* can vary between 0 and 8 Hz due to our reduced temporal sensitivity to faster repeat frequencies, while *Amplitude* is limited by the screen size). Thus, the bounded nature of the design space helps the HCI community understand the extent/bounds of what effects are possible.

The parametric nature of our framework is also beneficial when developing computer-aided design tools that provide objective design guidance on the quality of an interface design. This is because it is easier for the computer to understand parametric representations of highlighting techniques (i.e. in effect, each technique can be described as a vector of input parameter values) instead of dealing with imprecise natural language representations. Thus, it can be said that our framework is future-orientated in that it lays the foundations for the eventual development of such tools from the empirical data collected about different highlighting techniques.

There is however a risk that generational change in implementation technologies (e.g. a move away from screen/pixel-based displays of GUI's towards projected-light/holographic displays, or perhaps a shift towards increased use of physical objects/tokens [158]) may render parts of this framework obsolete. Specifically, since the units for many parameters are grounded in terms of pixels and other screen-based units (e.g. sRGB colour representations), these units may become outdated/irrelevant in such an environment. However, such large-scale changes would likely necessitate the development of a new/updated framework to provide coverage for the new possibilities opened up by the new developments.

9.2.2 Significance, Ecological Validity, and Limitations of Our Methods

The studies presented in this thesis confirm **H1.1**, demonstrating that it is possible to analyse the relative quality of highlighting techniques by concurrent measurement of their Noticeability and Distraction characteristics. This result is important, as it lays the foundations for further empirical studies of other highlighting techniques (using performance-based measures of their Noticeability and Distraction effects), by proving the feasibility of this approach. To our knowledge, no prior studies have managed to develop objective measures of both Noticeability and Distraction within the same experiment procedure.

Studies 1 and 2 present two different methodologies for conducting experiments to measure Noticeability and Distraction, covering two different use-cases for highlighting techniques:

- **Study 1** used an *Abstract Visual Search* task, using a novel method inspired by the Path Deviations paradigm. The purpose of this study was to understand basic perceptual factors influencing the strength of pop-out effects. A single-task design was used to increase the internal validity of the study, by ensuring that the experiment was only measuring the effects of the highlights, and not how well those highlights were able to capture participants' attention away from another task.

- **Study 2** used a *Dual Task* experiment methodology to be more ecologically valid than Study 1. It analysed Animated Window Border effects in a faux-desktop environment (whereas Study 1 analysed simple highlighting effects applied to a square-shaped grid of boxes instead), by using a dual-task experiment methodology (i.e. participants performed a dot-following task while responding to highlights when they appeared).

The tasks in both studies were relatively artificial. It is rare for users to perform the specific tasks used (e.g. dragging a target to a highlighted item, or performing a dot-following task). It is even rarer to be presented with a continuous stream of highlighting effects that they must react to rapidly. However, tradeoffs were required to yield an experiment method that facilitated the analysis of a large number of highlighting conditions under controlled conditions, while still using a task environment that sufficiently mimics the scenarios/use-cases that the experiment was designed to study.

9.2.3 Noticeability of Highlighting Techniques

The studies presented in this thesis attempted to measure the noticeability of highlighting techniques (i.e. how easy it was to detect the presence of those highlights) by using performance-based "Noticeability metrics". In both studies, the Noticeability metrics (N_T and N_t respectively) were obtained by measuring the amount of time taken by participants to respond to the highlighted item (i.e. in Study 1, participants dragged the cursor to the highlighted item, while in Study 2, participants pressed the spacebar when they detected the highlighted window). Overall, the metrics in both studies were successful at identifying significant differences between the performance of participants when exposed to different highlighting techniques. However, there were also some issues with these noticeability metrics related to the data collection methods used:

- **Study 1** – Targets moving rapidly (e.g. `HShake` or `LShake`) may have been difficult for participants to accurately target. Consequently, task times for these conditions may have been higher (leading to the N_T values to be lower, since Noticeability is inversely proportional to the time taken to notice a stimulus), as task time includes the pointing time in addition to the actual amount of time taken by participants to notice the highlight.
- **Study 2** – Ceiling effects limited the sensitivity of N_t, preventing it from being able to determine statistically-significant differences between the three most noticeable techniques (i.e. `Barberpole`, `FlashingLights`, and `GreenStripes`). However, subjective-experience measures for these techniques were able to distinguish between these techniques. This suggests that to increase the sensitivity of the experiment, it needs to be harder for participants to perform the task, allowing truly high-noticeability techniques to be more easily determined.

Despite the minor issues affecting some of the results, we were able to make some conclusions about the noticeability of different highlighting technique:

- Participants consider colour-based effects to be more noticeable than motion-based effects, despite performance-based metrics showing instead that some motion-based effects (e.g. in Study 1, `HPulse` and `LShake` were more noticeable than `HColor`).
- Artificial/Man-Made effects (e.g. `Barberpole`, `FlashingLights`, and `GreenStripes`) are more noticeable than naturally-inspired effects (e.g. `BigButterfly`, `SmallButterfly`, and `Fizzing`). This is consistent with our hypothesis (**H8.7**) that artificial or man-made techniques are more noticeable. A possible explanation for this is that people are more accustomed to ignoring nature-based stimuli (e.g. leaves and branches blowing in the wind, cloud movements, or butterflies flapping their wings). However, given that the "man-made" examples used in this study were inspired by effects used in traditional signage and advertising, it is also possible that these effects were particularly salient (e.g. due to years of evolutionary refinements) or that participants were already preconditioned to notice these.

9.3 Types of Distraction

At the start of this thesis, we defined "distraction" as follows:

> *Distraction refers to the undesired effects of a highlighting technique. These undesired effects may include performance degradation and annoyance.*

In light of the experiments conducted, we propose the following refined set of distraction types.

1. **Repeat Attention Capture** – The highlighted items cause distraction by repeatedly capturing attention, causing/forcing participants to repeatedly look at them.
2. **Misdirection of Attention** – Participants are initially misdirected to focus on the highlighted item instead of their real target (for example, because the highlight was more visually salient). Subsequently, task performance is affected because the diversion increased the total task time either directly (i.e. time spent focussing on the distractor) or indirectly (i.e. the distractor disrupted their scan pattern, making it harder to find their target). However, unlike Repeat Attention Capture, there is little continuing impact of the highlight on participant behaviour.
3. **Inability to Identify Known Target** – Participants are aware that something has been highlighted, but because the target is difficult to notice, they have to work hard to find the target. This affects performance in the primary task as participants are "distracted" from their primary task by this secondary (visual search) task that has temporarily captured their attention.

Further research is still needed to investigate whether there may other types of distraction that have not yet been discovered (or exposed by the work presented in this thesis). There are also opportunities to investigate how much else is currently unknown about the effects of different highlighting techniques, as well as what implications these findings will have for designers.

9.3.1 Repeat Attention Capture

Intuitively, **Repeat Attention Capture** is the most obvious type of distraction that often comes to mind. When running these experiments, our working hypothesis was that this was the way that traditionally "distracting" examples of highlighting techniques (e.g. flashing banner adverts, or some of the techniques ranked as being them most distracting in our two studies) worked. This was based on anecdotal experience that these techniques often feel distracting/annoying because they seem to be constantly pulling your attention back to them in the corner of your eye right beside some article you are trying to read, or making you think that "there is something there".

From the experiments conducted, we found that performance-based measures (in particular, those based on response time and/or motor-action behaviour) appear to be ineffective at measuring this type of distraction. It is possible that our motor control systems are sufficiently isolated from the cognitive systems that process and/or are affected by the negative effects of highlighting techniques, meaning that little if any reliable effects can be detected. This would explain why participants were still able to attain high levels of performance (i.e. "degraded" performance was often only slightly above baseline levels) for the "most distracting" techniques (as rated by the participants themselves) in the dot-following task. However, if this is the case, it would seemingly contradict the findings of the Path Deviations method used by Moher et al. [121].

This raises the question of whether eye tracking data could provide insights about this instead. The eye tracking data from Study 2 (Section 8.4.5) provides some weak evidence for the existence of this type of distraction: In Figure 8.23, the `Barberpole` and `FlashingLights` techniques (two of the most noticeable border types) show a modest fixation density on the candidate windows, suggesting that visual attention was drawn away from the primary task by those border effects. The density heatmaps for the `SmallButterfly` and `Glow` techniques can be interpreted in a similar way.

A limitation of using eye tracking data to measure Repeat Attention Capture is the assumption that participants they *repeatedly fixate* on the highlighted items. It ignores the possibility that participants may not need to actually fixate on the items at all (i.e. there are no eye movements to detect). Instead, mere awareness of the continuing presence of the highlight in peripheral vision, combined with the participant's prior mental map of that region of space [139] may be sufficient to cause the negative effects associated with Repeat Attention Capture.

This leads to the following promising direction of inquiry for future research:

Research Question (Future) 1
Perhaps Repeat Attention Capture happens when we constantly detect the presence of some salient stimuli in our peripheral vision, while knowing that said stimulus does not hold any value? The resulting cognitive dissonance (i.e. "there is something interesting to look at" versus "it is not actually worth looking at") is the likely source of the annoyance and frustration felt by the user, as they are presented with a constant stream of conflicting stimuli that they need to keep making a series of micro-decisions to ignore.

9.3.2 Misdirection of Attention

The second type of distraction – **Misdirection of Attention** – was the type of distraction measured in our first study (Chapter 7). In this experiment, the "distraction" characteristics of each highlighting technique were measured by applying the highlighting technique to an arbitrary item (crucially, anything that was *not* the target), and measuring how long it took participants to complete the task of finding the target item.

This type of distraction represents how distracting the highlighting effect would be if applied to the "wrong" items. Such a situation may occur if the designer's goals and the user's goals are misaligned. According to Don Norman's thesis that each system has three mental models associated with it (i.e. the "*Designer's (Intended) Model*", the "*System's (Projected) Model*", and the "*User's (Perceived) Model*") [126], such situations may happen more often than is intended. Thus, this measure of distraction is useful for designers for understanding the effects of a highlighting technique where there may be negative consequences for identifying the wrong target and/or also where speed of correctly detecting the target is important.

9.3.3 Inability to Identify Known Target

The third type of distraction – **Inability to Identify Known Target** – was the type of distraction measured in our second study (Chapter 8) by the Δd_{avg} metric. In this experiment, participants performed a dual task exercise in which their primary task was to perform a dot following task, and their secondary task was to respond when they had identified the peripheral window that was being highlighted. In addition to the highlighted windows, participants were also given a visual cue/hint that a highlighted window may be present. Thus, this experiment was an exercise in how well participants could identify a highlighting technique given that they knew that something was highlighted.

This type of distraction is effectively a measure of "**Inverse Noticeability**", or in other words, a measure of the *negative effects caused by the difficulty of noticing the highlighted item* given that participants knew that there was something highlighted that they should be attending to.

9.4 Future Work

The work presented in this thesis represents the first steps towards broadening and formalising our understanding of highlighting techniques. As our PCCH Design Framework (Chapter 4) and experiments have illustrated, there are many exciting directions for future research that have yet to be explored. This section identifies some promising and important directions for future research. These can be divided into several main categories: 1) Exploring variations on the experiment methodology to improve the sensitivity of the measures, 2) Performing studies to address knowledge gaps in the literature and/or to address interesting questions raised by the work presented here, and 3) Long-term goals for research on highlighting techniques.

9.4.1 Variations on These Experiment Methods

There are many opportunities to further develop these experiment methods, to improve on their limitations, and to investigate interesting variations of these techniques.

9.4.1.1 Interactions Between Highlighting Techniques

The studies presented in this thesis only measured the effects of single type of highlighting technique at a time (though there could be multiple instances of it in use). This was done to ensure that any effects detected were likely to have been caused by the technique being studied. However, highlighting techniques are not always used in isolation; sometimes multiple types of highlighting techniques may be used concurrently (e.g. two items in a 64-item grid are highlighted, one using `HPulse` and the other using `HShake`). This raises interesting questions about the ability of the user to detect a particular highlighting technique in the presence of other techniques.

To investigate, our methods could be modified such that multiple types of highlighting techniques are presented to participants in each trial. Participants would still be required to only respond to one type of highlighting technique. This leads to several possible variants:

1. There are two types of highlighting techniques shown. One of these changes in each trial, while the other one does not change and needs to be ignored.
2. There may be multiple types of highlighting techniques shown. Participants need to respond to the highlighted item with the correct marker or identifier (e.g. a letter/number).

9.4.1.2 Hybrid Methods

There are also opportunities to experiment with changing the tasks that participants needed to perform when responding to the highlighted item. Specifically, we wonder whether the problems affecting the noticeability measures could be avoided by swapping the way that participants responded to the target (highlighted item). That is,

- **Study 1** – Instead of pointing to the target, participants would only be required to tap the spacebar when they noticed a highlighted item. This variant of the procedure could mitigate the problem of the highlighted item being hard to click (e.g. for `HShake` and `LPulse`) due to the target's position on screen changing faster than participants can track it.
- **Study 2** – Instead of tapping the spacebar, participants would be required to click on the highlighted window. This would force them to actually identify which window was highlighted, instead of being able to rely on their peripheral vision only. Combined with the time required to point to the target, this may be enough to increase the sensitivity of the measure by slowing down the speed at which participants can respond.

If both of these tweaks are successful, it would suggest that both methods for measuring noticeability (i.e. pressing a key, and pointing to the target) need to be included in a study, if accurate measurement of the effects are desired. That is, the key-press variant would offer the best approximation of when the highlights are first detected, while the pointing variant could be useful in cases where there is a ceiling effect limiting the sensitivity of the method. However, this two-in-one method was not used because it would have increased the length of the experiment (or alternatively, reduced the number of conditions that could be covered); it would also have been harder to explain to participants what they needed to do.

9.4.2 Interesting Directions to Investigate

In addition to improving on the methods presented in this thesis, there are many interesting directions and questions for future research to address.

9.4.2.1 Sampling the Design Space

The studies presented in this thesis were only "proof-of-concept" studies of the viability of the experiment protocols for measuring the effects of highlighting techniques. Thus, we deliberately chose example techniques that were more representative of "extreme" datapoints, so that it would be easier to determine if the measures were detecting any effects. However, as shown in Chapter 4, the design space for highlighting techniques is very broad and varied; as a result, there is still a great need for further research into the effects of different combination of parameters. These efforts can be divided into two broad categories of work:

1. **Depth: Sample the Full Parameter Space for Each Technique** – There is currently a lack of knowledge about the shape (or response curves) of the design space. The work in this thesis presented a few point-samples (i.e. at two extreme points in Study 1, and a isolated points in Study 2); however, there is still a need for denser sampling of the range for each parameter (e.g. Frequency and Amplitude, or Colour Intensity/Contrast) and combinations thereof, to better understand the subtle (likely non-linear) relationships between different combinations of parameter strengths. For example, this could allow designers to understand how similar level of Noticeability or

Distraction could be achieved by using a lower frequency combined with a higher amplitude.

2. **Breadth: Study Different Types of Highlighting Techniques** – The work presented in this thesis (and that of the prior literature) has only covered a small number of highlighting techniques. However, as illustrated by our design framework (to a lesser extent, Study 2), there are many possibilities for constructing highlighting techniques, many of which have not been studied in great depth yet. Thus, there are opportunities to study the effects of these techniques.

9.4.2.2 Multi-State Highlighting Techniques

The highlighting techniques studied in this thesis have all been simple two-state techniques (i.e. *Normal* (No Highlighting) and *Highlighted*). However, as discussed in Section 4.4.1, highlighting techniques can be constructed with several other states (i.e. *Initial Onset*, *Dismissal*, and *Reminder*). Thus, an interesting direction for future research could be to investigate the relative merits of highlighting techniques that use these other states. This would likely require finding a suitable task environment where these techniques are most beneficial, and developing an experimental protocol that operationalises this task to evaluate the effectiveness of these highlighting techniques.

9.4.2.3 Environmental Concerns

The studies presented in this thesis assumed that it was unlikely that participants modified their behaviour patterns during the course of each session in response to the stimuli they had seen. That is, there were negligible changes to the implicit Payoff Matrices that each participant had for how they respond the highlights, based on the relative costs and benefits to adopting different strategies (i.e. Strict/Risk-Adverse or Loose/Loss-Adverse).

This raises some interesting questions: 1) Is it possible to manipulate the strategies people use for detecting and responding to highlighting techniques, and 2) What are the consequences of doing so.

9.4.2.4 Other Metrics

This thesis has focussed on using Noticeability and Distraction as metrics of highlighting technique effectiveness. However, there are other other metrics that could also be used (e.g. Accuracy of detection, or ability to correctly identify the highlighted item, or True Positive Rate (TPR) and False Positive Rate (FPR)). There are opportunities to further investigate the utility of these other metrics. There are also interesting questions about how these alternative metrics correspond with Noticeability and Distraction.

Another interesting direction is to develop an "*Effectiveness*" metric that provides a score on the overall *goodness* of a technique based on its Noticeability and Distraction metrics. It would be like a "signal-to-noise" ratio for highlighting techniques, and would likely make it easier for designers to quickly compare two techniques as they only have to interpret a single measure.

9.4.2.5 Questions Raised by Subjective Experience

The subjective experience questions exposed a few promising insights that raise some interesting questions that future research should investigate:

- **Initial Preference versus Repeat Exposure** – The subjective feedback received in Study 2 suggests that participants' perceptions of the techniques changed over the course of the experiment. Specifically, they found techniques (e.g. `BigButterfly`) quite "novel" and "creative" when they first encountered those techniques; however, they also quickly found (i.e. after 1-2 blocks of trials or exposures to the effect) that such effects were quite distracting when they appear in the middle of a task.
- **Attractiveness versus Annoyance** - It would be interesting to investigate whether there is any link between how visually attractive participants find a highlighting technique, and how annoying the effect is. Common sense would suggest that less visually attractive techniques are less likely to be considered to be annoying.

9.4.3 Development of Predictive Models and Design Tools

Ultimately, studies of highlighting technique effects all contribute toward the goal of improving of understanding of highlighting techniques, which is a necessary prerequisite for providing useful design guidance during the design process. However, as noted by Rosenholtz et al. [140], simply providing guidance in the form "design guidelines" may be inadequate. This is because human designers may not be able to always fully understand and consistently apply all the design guidelines, while taking care of any subtle interactions between certain combinations of guidelines.

Computer Aided Design Tools (CADT's) are a promising solution for providing design guidance in and objective and reliable manner while taking into account the full complexity of the domain. CADT's allow designers to experiment with how different designs may perform without needing to run user studies to find out. Instead, they work by taking the prototype designs that the designer is developing, and running various analysis tools on the prototype. Analyses which can be performed include simulations of user behaviour (e.g. CogTool [152, 31] and QN-MHP [109, 175]), computation of metrics/salience maps (e.g. iNVT [90, 89], DesignEye [140], and Complexity/Colourfulness Metrics [137, 82]), and heuristic analyses (e.g. [99]). The results of these analyses are then shown to the designer, providing design teams with objective feedback that they can incorporate into their design process.

The development of useful CADT's for highlighting techniques requires the availability of comprehensive datasets of the effects of highlighting techniques (e.g. Noticeability and Distraction) relative to the HL Parameters used to achieve that result. With such a dataset, several different approaches could be used to develop the analysis tools (i.e. computational models) used to power a CADT:

- **Machine Learning** – One approach would be to apply popular machine-learning techniques on such a dataset. This would involve using the results of prior studies as "training data". By using a machine-friendly classification system based on PCCH (Chapter 4), it becomes easier to treat the HL Parameters defining each technique as a vector of inputs/variables to the system.
- **Calibration of Monte-Carlo Simulations** – Another approach would be to develop a Discrete Event Simulation [127] model of how people interact with highlighting techniques, and using empirical data to calibrate the weights on various random processes (e.g. the tendency of the user to focus on the most salient item versus focussing on a random location instead). Such a simulation model could make use of prior models such as iNVT [89] and QN-MHP [109] to provide a realistic simulation of where visual attention would be directed under optimal conditions.

10 Conclusion

Highlighting techniques are visual communication techniques that make users aware of salient information in a timely manner. They are commonly used across a wide range of different interfaces, for many purposes, and in many forms. However, there has traditionally been a lack of understanding about how to select, apply, and control their effects to obtain the best balance between the need to be noticeable while reducing unnecessary distraction.

This thesis represents a step towards resolving this knowledge gap. The work presented provides the following primary contributions to knowledge about highlighting techniques, and how their effects can be analysed and compared:

- A review and summary of foundational insights (Chapter 2) and prior studies (Chapter 3) of highlighting techniques/effects across the HCI and Psychology literature.
- The development a structured framework for describing/modelling highlighting techniques in terms of how they are constructed and controlled (Chapter 4).
- The development a new empirical framework for measuring and comparing the relative quality of highlighting techniques in terms of their emergent Noticeability and Distraction characteristics (Chapter 6).
- The development and evaluation of two experiment protocols for measuring the Noticeability and Distraction effects of highlighting techniques:
    1. The first method (Chapter 7) was a proof-of-concept demonstration that the Noticeability and Distraction characteristics of highlighting techniques can be measured using performance based metrics. It uses an abstract visual search task, where participants dragged a marker to a target item within a grid of 16 or 64 items while being exposed to different configurations of 4 commonly used highlighting techniques (each at 2 levels of intensity).
    2. The second method (Chapter 8) was a demonstration that Noticeability and Distraction can also be measured in more realistic scenarios. The Noticeability and Distraction characteristics of 8 animated window border effects applied to a mock desktop environment were measured using a dual-task paradigm, where participants had to follow the movements of a dot moving around a cog-shaped path while tapping the spacebar in response to windows getting highlighted.
- A refined understanding of the nature of "distraction" as it applies to highlighting techniques. Specificially, we propose that there are at least 3 types of distraction effects that the user may experience when interacting with highlighting techniques: Repeat Attention Capture, Misdirection of Attention, and Inability to Identify Known Target.

This research provides the foundations for conducting future research on analysing and comparing the effects of highlighting techniques in a more systematic and structured way. There are many opportunities for future research to use, extend, and/or further develop the methods presented in this thesis. These include using our experiment protocols to characterise techniques across the highlighting design space, and developing new experiment methods to investigate promising directions of inquiry identified/exposed by our work.

Bibliography

[1] Web Content Accessibility Guidelines (WCAG) 2.0 – General Flash and Red Flash Thresholds. URL: `http://www.w3.org/TR/2008/REC-WCAG20-20081211/#general-thresholddef`, 2008.

[2] Web Content Accessibility Guidelines (WCAG) 2.0 – Guideline 2.3 Seizures: Do not design content in a way that is known to cause seizures. URL: `http://www.w3.org/TR/WCAG20/#seizure`, 2008.

[3] Scalable Vector Graphics (SVG) 1.1 (Second Edition) – Recognized color keyword names. URL: `https://www.w3.org/TR/SVG/types.html#ColorKeywords`, 2011.

[4] Web Content Accessiblity Guidelines (WCAG) 2.0 – Pause Stop Hide. URL: `http://www.w3.org/TR/UNDERSTANDING-WCAG20/time-limits-pause.html`, 2015.

[5] Tobii Technology AB. *User Manual – Tobii Studio*, version 3.2 rev a edition, 11 2012.

[6] Johnny Accot and Shumin Zhai. Beyond Fitts' Law: Models for Trajectory-based HCI Tasks. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, pages 295–302, New York, NY, USA, 1997. ACM.

[7] Piotr D. Adamczyk and Brian P. Bailey. If Not Now, when?: The Effects of Interruption at Different Moments Within Task Execution. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 271–278, New York, NY, USA, 2004. ACM.

[8] Jack A Adams. A closed-loop theory of motor learning. *Journal of motor behavior*, 3(2):111–150, 1971.

[9] Edward H. Adelson. Checkerboard Shadow Illusion, 1995.

[10] Elena Agapie, Gene Golovchinsky, and Pernilla Qvarfordt. Leading People to Longer Queries. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 3019–3022, New York, NY, USA, 2013. ACM.

[11] Jason Alexander, Andy Cockburn, Stephen Fitchett, Carl Gutwin, and Saul Greenberg. Revisiting read wear: analysis, design, and evaluation of a footprints scrollbar. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1665–1674, New York, NY, USA, 2009. ACM.

[12] Bonnie Brinton Anderson, C. Brock Kirwan, Jeffrey L. Jenkins, David Eargle, Seth Howard, and Anthony Vance. How Polymorphic Warnings Reduce Habituation in the Brain: Insights from an fMRI Study. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 2883–2892, New York, NY, USA, 2015. ACM.

[13] Matthew Anderson, Ricardo Motta, Srinivasan Chandrasekar, and Michael Stokes. Proposal For A Standard Default Color Space For The Internet – sRGB. In *Color and imaging conference*, volume 1996, pages 238–245. Society for Imaging Science and Technology, 1996.

[14] Apple. macOS Human Interface Guidelines – Animation. URL: https://developer.apple.com/library/content/documentation/UserExperience/Conceptual/OSXHIGuidelines/Animation.html, October 2016.

[15] Larry Arend, Alex Logan, and Galina Havin. Blinking, Flashing, and Temporal Response. URL: http://colorusage.arc.nasa.gov/flashing_2.php, 2003.

[16] Larry Arend, Alex Logan, and Galina Havin. Color Discrimination and Identification. URL: http://colorusage.arc.nasa.gov/discrim.php, 2003.

[17] Larry Arend, Alex Logan, and Galina Havin. Color Guidelines Hierarchy. URL: http://colorusage.arc.nasa.gov/GuidelinesHierarchy.php, 2003.

[18] Larry Arend, Alex Logan, and Galina Havin. Creating Perceptual Layers. URL: http://colorusage.arc.nasa.gov/Percept_layers_1.php, 2003.

[19] Larry Arend, Alex Logan, and Galina Havin. Designing With Blue. URL: http://colorusage.arc.nasa.gov/blue_2.php, 2003.

[20] Larry Arend, Alex Logan, and Galina Havin. Designing with Luminance Contrast. URL: http://colorusage.arc.nasa.gov/design_lum_0.php, 2003.

[21] Larry Arend, Alex Logan, and Galina Havin. Guidelines for Color Discrimination and Identification. URL: http://colorusage.arc.nasa.gov/guidelines_discrim_id.php, 2003.

[22] Larry Arend, Alex Logan, and Galina Havin. Labelling with Color. URL: http://colorusage.arc.nasa.gov/labeling.php, 2003.

[23] Larry Arend, Alex Logan, and Galina Havin. Legibility. URL: http://colorusage.arc.nasa.gov/legib_1.php, 2003.

[24] Larry Arend, Alex Logan, and Galina Havin. Luminance and Chromaticity. URL: https://colorusage.arc.nasa.gov/lum_and_chrom.php, 2003.

[25] Jeff Atwood. Complaint Driven Design. URL: http://www.codinghorror.com/blog/2014/02/complaint-driven-development.html, 2014.

[26] Autodesk. What's New: Highlight What's New — Maya. URL: https://knowledge.autodesk.com/support/maya/learn-explore/caas/CloudHelp/cloudhelp/2015/ENU/Maya/files/GUID-84AF4FA6-BE21-4F4E-ACCF-7AD005E13EBF-htm.html, 2015.

[27] Brian P. Bailey and Joseph A. Konstan. On the need for attention-aware systems: Measuring effects of interruption on task performance, error rate, and affective state. *Computers in Human Behavior*, 22(4):685 – 708, 2006. Attention aware systems Special issue: Attention aware systems.

[28] Linda Ruth Bartram. *Enhancing information visualization with motion*. PhD thesis, Simon Fraser University, 2001.

[29] Lyn Bartram, Colin Ware, and Tom Calvert. Moticons: Detection, Distraction and Task. *Int. J. Hum.-Comput. Stud.*, 58(5):515–545, May 2003.

[30] Ashweeni Kumar Beeharee, Adrian J. West, and Roger Hubbold. Visual Attention Based Information Culling for Distributed Virtual Environments. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, VRST '03, pages 213–222, New York, NY, USA, 2003. ACM.

[31] R. Bellamy, B. John, and S. Kogan. Deploying CogTool: Integrating Quantitative Usability Assessment Into Real-World Software Development. In *Software Engineering (ICSE), 2011 33rd International Conference on*, pages 691–700, May 2011.

[32] Jan Panero Benway and David M. Lane. Banner Blindness: Web Searchers Often Miss "Obvious" Links. URL `http://www.internettg.org/newsletter/dec98/banner_blindness.html`, December 1998.

[33] Ofer Bergman, Ruth Beyth-Marom, Rafi Nachmias, Noa Gradovitch, and Steve Whittaker. Improved search engines and navigation preference in personal information management. *ACM Trans. Inf. Syst.*, 26(4):20:1–20:24, October 2008.

[34] Ofer Bergman, Oded Elyada, Noga Dvir, Yana Vaitzman, and Adir Ben Ami. Spotting the Latest Version of a File with Old'nGray. *Interacting with Computers*, 2014.

[35] Ofer Bergman, Maskit Tene-Rubinstein, and Jonathan Shalom. The use of attention resources in navigation versus search. *Personal and Ubiquitous Computing*, 17(3):583–590, 2013.

[36] Ofer Bergman, Steve Whittaker, Mark Sanderson, Rafi Nachmias, and Anand Ramamoorthy. The effect of folder structure on personal file navigation. *J. Am. Soc. Inf. Sci. Technol.*, 61(12):2426–2441, December 2010.

[37] Jacques Bertin. *Semiology of Graphics*. University of Wisconsin Press, 1983.

[38] Meera M. Blattner, Denise A. Sumikawa, and Robert M. Greenberg. Earcons and Icons: Their Structure and Common Design Principles. *Hum.-Comput. Interact.*, 4(1):11–44, March 1989.

[39] Kenneth R Boff, Lloyd Kaufman, and James P Thomas. Handbook of perception and human performance. 1986.

[40] Ali Borji and Laurent Itti. State-of-the-art in visual attention modeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):185–207, 2013.

[41] Herman Bouma. Interaction effects in parafoveal letter recognition. *Nature*, 226:177–178, 1970.

[42] Pascal Brachet. Texmaker – The Universal LaTeX Editor. URL: `http://www.xm1math.net/texmaker/`, 2017.

[43] Moira Burke, Nicholas Gorman, Erik Nilsen, and Anthony Hornof. Banner Ads Hinder Visual Search and Are Forgotten. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '04, pages 1139–1142, New York, NY, USA, 2004. ACM.

[44] Stuart K Card, Thomas P Moran, and Allen Newell. The Model Human Processor: An engineering model of human performance. *Handbook of Human Perception*, 2, 1986.

[45] MST Carpendale. Considering visual variables as a basis for information visualisation. 2003.

[46] Edwin Catmull and Raphael Rom. A class of local interpolating splines. *Computer aided geometric design*, 74:317–326, 1974.

[47] Bay-Wei Chang and David Ungar. Animation: From Cartoons to the User Interface. Technical report, Mountain View, CA, USA, 1995.

[48] Kuang Chen, Joseph M Hellerstein, and Tapan S Parikh. Designing adaptive feedback for improving data entry accuracy. In *Proceedings of the 23nd annual ACM Symposium on User Interface Software and Technology*, pages 239–248. ACM, 2010.

[49] Fanny Chevalier, Pierre Dragicevic, Anastasia Bezerianos, and Jean-Daniel Fekete. Using text animated transitions to support navigation in document histories. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 683–692. ACM, 2010.

[50] Richard Chimera. Value Bars: An information visualization and navigation tool for multi-attribute listings. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 293–294. ACM, 1992.

[51] Andy Cockburn, Carl Gutwin, and Saul Greenberg. A Predictive Model of Menu Performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 627–636, New York, NY, USA, 2007. ACM.

[52] The Qt Company. QEasingCurve. URL: `http://doc.qt.io/qt-5.6/qeasingcurve.html`, 2016.

[53] The Qt Company. QML Applications. URL: `http://doc.qt.io/qt-5/qmlapplications.html`, 2017.

[54] Edward Cutrell, Mary Czerwinski, and Eric Horvitz. Notification, Disruption, and Memory: Effects of Messaging Interruptions on Memory and Performance. In *Human-Computer Interaction–Interact '01*, pages 263–269. IOS Press, 2001.

[55] Dianne Cyr, Milena Head, and Hector Larios. Colour Appeal in Website Design Within and Across Cultures: A Multi-method Evaluation. *Int. J. Hum.-Comput. Stud.*, 68(1-2):1–21, January 2010.

[56] Laura Dabbish, Gloria Mark, and Víctor M. González. Why Do I Keep Interrupting Myself?: Environment, Habit and Self-interruption. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 3127–3130, New York, NY, USA, 2011. ACM.

[57] Thomas Davies and Ashweeni Beeharee. The Case of the Missed Icon: Change Blindness on Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1451–1460, New York, NY, USA, 2012. ACM.

[58] Linux Mint Developers. Introducing Nemo. URL: `http://cinnamon.linuxmint.com/?p=198`, September 2012.

[59] Pierre Dragicevic, Anastasia Bezerianos, Waqas Javed, Niklas Elmqvist, and Jean-Daniel Fekete. Temporal Distortion for Animated Transitions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2009–2018, New York, NY, USA, 2011. ACM.

[60] Pierre Dragicevic, Stéphane Huot, and Fanny Chevalier. Gliimpse: Animating from markup code to rendered documents and vice versa. In *Proceedings of the 24th annual ACM symposium on User Interface Software and Technology*, pages 257–262. ACM, 2011.

[61] A. Duchowski. *Eye Tracking Methodology: Theory and Practice*. Methods in molecular biology. Springer London, 2007.

[62] Mario J. Enriquez and Karon E. MacLean. The Hapticon Editor: A Tool in Support of Haptic Communication Research. In *Proceedings of the 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS'03)*, HAPTICS '03, pages 356–, Washington, DC, USA, 2003. IEEE Computer Society.

[63] David Erceg. Explorer++. URL: `http://explorerplusplus.com/`, 2013.

[64] Leah Findlater, Karyn Moffatt, Joanna McGrenere, and Jessica Dawson. Ephemeral Adaptation: The use of gradual onset to improve menu selection performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1655–1664. ACM, 2009.

[65] Jason Fischer and David Whitney. Serial dependence in visual perception. *Nature neuroscience*, 17(5):738–743, 2014.

[66] Stephen Fitchett. *Understanding and Improving Personal File Retrieval*. PhD thesis, University of Canterbury, 2014.

[67] Stephen Fitchett, Andy Cockburn, and Carl Gutwin. Improving navigation-based file retrieval. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2329–2338, New York, NY, USA, 2013. ACM.

[68] Paul M Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, 47(6):381, 1954.

[69] James Fogarty, Scott E. Hudson, Christopher G. Atkeson, Daniel Avrahami, Jodi Forlizzi, Sara Kiesler, Johnny C. Lee, and Jie Yang. Predicting Human Interruptibility with Sensors. *ACM Trans. Comput.-Hum. Interact.*, 12(1):119–146, March 2005.

[70] James Foley, Steven K. Feiner, and John F Hughes. *Computer Graphics: Principles and Practice*, page 1174. Addison-Wesley, Reading, MA, USA, 1990.

[71] George Furnas. Generalized Fisheye Views. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '86, pages 16–23, New York, NY, USA, 1986. ACM.

[72] Jason P Gallivan and Craig Stuart Chapman. Three-dimensional reach trajectories as a probe of real-time decision-making between multiple competing targets. *Frontiers in Neuroscience*, 8(215), 2014.

[73] Emmanouil Giannisakis, Gilles Bailly, Sylvain Malacria, and Fanny Chevalier. IconHK: Using Toolbar Button Icons to Communicate Keyboard Shortcuts. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 4715–4726, New York, NY, USA, 2017. ACM.

[74] Jennifer Gluck, Andrea Bunt, and Joanna McGrenere. Matching Attentional Draw with Utility in Interruption. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 41–50, New York, NY, USA, 2007. ACM.

[75] Gnome. Nautilis File Browser. URL: `https://wiki.gnome.org/action/show/Apps/Nautilus?action=show&redirect=Nautilus`, 2013.

[76] Google. *Material Design – Material Motion*, 2016. Last Updated: 7 September 2016.

[77] Google. Chrome Web Browser. URL: `http://www.google.com/chrome`, 2017.

[78] Tovi Grossman, Pierre Dragicevic, and Ravin Balakrishnan. Strategies for accelerating on-line learning of hotkeys. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1591–1600. ACM, 2007.

[79] Carl Gutwin, Andy Cockburn, and Ashley Coveney. Peripheral Popout: The Influence of Visual Angle and Stimulus Intensity on Popout Effects. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 208–219, New York, NY, USA, 2017. ACM.

[80] Chris Harrison, John Horstman, Gary Hsieh, and Scott Hudson. Unlocking the expressivity of point lights. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, pages 1683–1692. ACM, 2012.

[81] Chris Harrison, Gary Hsieh, Karl DD Willis, Jodi Forlizzi, and Scott E Hudson. Kineticons: Using iconographic motion in graphical user interface design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1999–2008. ACM, 2011.

[82] David Hasler and Sabine E Suesstrunk. Measuring colorfulness in natural images. In *Electronic Imaging 2003*, pages 87–95. International Society for Optics and Photonics, 2003.

[83] Edmund Heery and Mike Noon. Central Tendency Bias. URL: `http://www.oxfordreference.com/10.1093/oi/authority.20110803095558737`.

[84] William C. Hill, James D. Hollan, Dave Wroblewski, and Tim McCandless. Edit wear and read wear. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '92, pages 3–9, New York, NY, USA, 1992. ACM.

[85] James E Hoffman. Visual attention and eye movements. *Attention*, 31:119–153, 1998.

[86] Anthony J. Hornof, Yunfeng Zhang, and Tim Halverson. Knowing Where and When to Look in a Time-critical Multimodal Dual Task. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 2103–2112, New York, NY, USA, 2010. ACM.

[87] Scott E Hudson and John T Stasko. Animation support in a user interface toolkit: Flexible, robust, and reusable abstractions. In *Proceedings of the 6th annual ACM symposium on User interface software and technology*, pages 57–67. ACM, 1993.

[88] Shamsi T. Iqbal and Brian P. Bailey. Investigating the Effectiveness of Mental Workload As a Predictor of Opportune Moments for Interruption. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, pages 1489–1492, New York, NY, USA, 2005. ACM.

[89] Laurent Itti, Nitin Dhavale, and Frederic Pighin. Realistic avatar eye and head animation using a neurobiological model of visual attention. In *Optical Science and Technology, SPIE's 48th Annual Meeting*, pages 64–78. International Society for Optics and Photonics, 2004.

[90] Laurent Itti and Christof Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision research*, 40(10):1489–1506, 2000.

[91] Rob Jacob and Sophie Stellmach. What You Look At Is What You Get: Gaze-Based User Interfaces. *Interactions*, 28, August 2016.

[92] Brian Johnson and Ben Shneiderman. Tree-Maps: a space-filling approach to the visualization of hierarchical information structures. In *Proceedings of the 2nd conference on Visualization '91*, VIS '91, pages 284–291, Los Alamitos, CA, USA, 1991. IEEE Computer Society Press.

[93] Jeff Johnson. *GUI Bloopers: Don'ts and Do's for software developers and Web designers*. Morgan Kaufmann, 2000.

[94] Jeff Johnson. *Designing with the mind in mind: Simple guide to understanding user interface design rules*. Morgan Kaufmann, 2010.

[95] Daniel Kahneman. *Attention and Effort*. Prentice-Hall, Englewood Cliffs, New Jersey, 1973.

[96] Rubaiat Habib Kazi, Tovi Grossman, Nobuyuki Umetani, and George Fitzmaurice. SKUID: Sketching Stylized Animated Drawings with Motion Amplifiers. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '16, pages 6–6, New York, NY, USA, 2016. ACM.

[97] KDE. Dolphin File Manager. URL: `http://dolphin.kde.org/`, 2013.

[98] Michael Kerrisk. diff(1) – Linux manual page. URL: `http://man7.org/linux/man-pages/man1/diff.1.html`, May 2017.

[99] David E. Kieras and Anthony J. Hornof. Towards Accurate and Practical Predictive Models of Active-Vision-based Visual Search. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pages 3875–3884, New York, NY, USA, 2014. ACM.

[100] Ji Hye Kim and Kun Pyo Lee. Cultural Difference and Mobile Phone Interface Design: Icon Recognition According to Level of Abstraction. In *Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices &Amp; Services*, MobileHCI '05, pages 307–310, New York, NY, USA, 2005. ACM.

[101] RT Knight, MF Grabowecky, and D Scabini. Role of Human Prefrontal Cortex in Attention Control. *Advances in neurology*, 66:21, 1995.

[102] Kurt Koffka. Principles of Gestalt psychology. 1935.

[103] Steve Krug. *Don't Make Me Think: A Common Sense Approach to the Web (2nd Edition)*. New Riders Publishing, Thousand Oaks, CA, USA, 2005.

[104] John Lasseter. Principles of Traditional Animation Applied to 3D Computer Animation. *SIGGRAPH Comput. Graph.*, 21(4):35–44, August 1987.

[105] Joshua Leung. Animato – An Improved Animation System for Blender. URL: `https://wiki.blender.org/index.php/Dev:2.5/Source/Animation/Animato`, January 2009.

[106] Jie Liang and Mao Lin Huang. Highlighting in information visualization: A survey. In *Information Visualisation (IV), 2010 14th International Conference*, pages 79–85. IEEE, 2010.

[107] Gitte Lindgaard, Cathy Dudek, Devjani Sen, Livia Sumegi, and Patrick Noonan. An Exploration of Relations Between Visual Appeal, Trustworthiness and Perceived Usability of Homepages. *ACM Trans. Comput.-Hum. Interact.*, 18(1):1:1–1:30, May 2011.

[108] Gitte Lindgaard, Gary Fernandes, Cathy Dudek, and Judith Brown. Attention web designers: You have 50 milliseconds to make a good first impression! *Behaviour & information technology*, 25(2):115–126, 2006.

[109] Yili Liu, Robert Feyen, and Omer Tsimhoni. Queueing Network-Model Human Processor (QN-MHP): A Computational Architecture for Multitask Performance in Human-machine Systems. *ACM Trans. Comput.-Hum. Interact.*, 13(1):37–70, March 2006.

[110] Alan M MacEachren. *How maps work: representation, visualization, and design*. Guilford Press, 1995.

[111] Paul P Maglio and Christopher S Campbell. Tradeoffs in displaying peripheral information. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 241–248. ACM, 2000.

[112] Sylvain Malacria, Gilles Bailly, Joel Harrison, Andy Cockburn, and Carl Gutwin. Promoting Hotkey Use Through Rehearsal with ExposeHK. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 573–582, New York, NY, USA, 2013. ACM.

[113] Gabriela Mancero, William Wong, and Paola Amaldi. Looking but Not Seeing: Implications for HCI. In *Proceedings of the 14th European Conference on Cognitive Ergonomics: Invent! Explore!*, ECCE '07, pages 167–174, New York, NY, USA, 2007. ACM.

[114] Gloria Mark, Daniela Gudith, and Ulrich Klocke. The Cost of Interrupted Work: More Speed and Stress. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 107–110, New York, NY, USA, 2008. ACM.

[115] Colin Martindale, Kathleen Moore, and Jonathan Borkum. Aesthetic Preference: Anomalous findings for Berlyne's psychobiological theory. *The American Journal of Psychology*, pages 53–80, 1990.

[116] Justin Matejka, Tovi Grossman, and George Fitzmaurice. Ambient Help. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2751–2760, New York, NY, USA, 2011. ACM.

[117] Justin Matejka, Tovi Grossman, and George Fitzmaurice. Patina: Dynamic Heatmaps for Visualizing Application Usage. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 3227–3236, New York, NY, USA, 2013. ACM.

[118] D Scott McCrickard, Richard Catrambone, Christa M Chewar, and John T Stasko. Establishing tradeoffs that leverage attention for utility: Empirically evaluating information display in notification systems. *International Journal of Human-Computer Studies*, 58(5):547–582, 2003.

[119] D. Scott McCrickard, C. M. Chewar, Jacob P. Somervell, and Ali Ndiwalana. A Model for Notification Systems Evaluation – Assessing User Goals for Multitasking Activity. *ACM Trans. Comput.-Hum. Interact.*, 10(4):312–338, December 2003.

[120] D Scott McCrickard and Christa M Chewar. Attuning notification design to user goals and attention costs. *Communications of the ACM*, 46(3):67–72, 2003.

[121] Jeff Moher, Brian A Anderson, and Joo-Hyun Song. Dissociable effects of salience on attention and goal-directed action. *Current Biology*, 25(15):2040–2046, 2015.

[122] Kenneth Moreland. Diverging Color Maps for Scientific Visualization. In *Proceedings of the 5th International Symposium on Advances in Visual Computing: Part II*, ISVC '09, pages 92–103, Berlin, Heidelberg, 2009. Springer-Verlag.

[123] GA Moss and RW Gunn. Gender Differences in Website Production and Preference Aesthetics: preliminary implications for ICT in education and beyond. *Behaviour & Information Technology*, 28(5):447–460, 2009.

[124] Heiko Müller, Martin Pielot, and Rodrigo de Oliveira. Towards Ambient Notifications. *Peripheral Interaction: Embedding HCI in Everyday Life*, page 21, 2013.

[125] Mozilla Developer Network. Frame rate – Firefox Developer Tools — MDN. URL: `https://developer.mozilla.org/en-US/docs/Tools/Performance/Frame_rate`, 2015.

[126] Donald A. Norman. *The Design of Everyday Things*. Doubleday Business, 1990.

[127] Bernd Page, Wolfgang Kreutzer, and Björn Gehlsen. *The Java simulation handbook: Simulating discrete event systems with UML and Java*. Shaker Verlag, 2005.

[128] Robert Penner. Robert Penner Easing Equations. URL: `http://www.robertpenner.com/easing/`, 2001.

[129] Robert Penner. *Robert Penner's Programming Macromedia Flash MX*. McGraw-Hill, Inc., 2002.

[130] Ken. Perlin. Real Time Responsive Animation With Personality. *Visualization and Computer Graphics, IEEE Transactions on*, 1(1):5–15, Mar 1995.

[131] Peter Pirolli and Stuart Card. Information Foraging. *Psychological Review*, 106(4):634–675, 1999.

[132] Philip Quinn, Andy Cockburn, Indratmo, and Carl Gutwin. An Investigation of Dynamic Landmarking Functions. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '08, pages 322–325, New York, NY, USA, 2008. ACM.

[133] Jeff Raskin. *The Humane Interface: New Directions for Designing Interactive Systems*. Addison-Wesley, 2000.

[134] William T Reeves. Particle Systems – A Technique for Modeling a Class of Fuzzy Objects. *ACM Transactions on Graphics (TOG)*, 2(2):91–108, 1983.

[135] William T. Reeves, Eben F. Ostby, and Samuel J. Leffler. The Menv Modelling and Animation Environment. *The Journal of Visualization and Computer Animation*, 1(1):33–40, 1990.

[136] Katharina Reinecke and Abraham Bernstein. Improving Performance, Perceived Usability, and Aesthetics with Culturally Adaptive User Interfaces. *ACM Trans. Comput.-Hum. Interact.*, 18(2):8:1–8:29, July 2011.

[137] Katharina Reinecke and Krzysztof Z. Gajos. Quantifying Visual Preferences Around the World. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 11–20, New York, NY, USA, 2014. ACM.

[138] Katharina Reinecke, Tom Yeh, Luke Miratrix, Rahmatri Mardiko, Yuechen Zhao, Jenny Liu, and Krzysztof Z. Gajos. Predicting Users' First Impressions of Website Aesthetics with a Quantification of Perceived Visual Complexity and Colorfulness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2049–2058, New York, NY, USA, 2013. ACM.

[139] Ronald A Rensink. Change Blindness. *Neurobiology of Attention*, pages 76–81, 2005.

[140] Ruth Rosenholtz, Amal Dorai, and Rosalind Freeman. Do Predictions of Visual Perception Aid Design? *ACM Trans. Appl. Percept.*, 8(2):12:1–12:20, February 2011.

[141] Ruth Rosenholtz, Jie Huang, Alvin Raj, Benjamin J. Balas, and Livia Ilie. A summary statistic representation in peripheral vision explains visual search. *Journal of Vision*, 12(4):14, 2012.

[142] Ruth Rosenholtz, Yuanzhen Li, and Lisa Nakano. Measuring Visual Clutter. *Journal of vision*, 7(2):17, 2007.

[143] Joey Scarr. *Understanding and Exploiting Spatial Memory in the Design of Efficient Command Selection Interfaces.* PhD thesis, University of Canterbury, 2014.

[144] Joey Scarr, Andy Cockburn, Carl Gutwin, and Andrea Bunt. Improving command selection with CommandMaps. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 257–266, New York, NY, USA, 2012. ACM.

[145] Joey Scarr, Andy Cockburn, Carl Gutwin, and Sylvain Malacria. Testing the robustness and performance of spatially consistent interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 3139–3148, New York, NY, USA, 2013. ACM.

[146] Brad Shaw. CSS3 = Awesome. URL: `http://css3.bradshawenterprises.com/`, 2013.

[147] Dave Shreiner and The Khronos OpenGL ARB Working Group. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Versions 3.0 and 3.1*. Addison-Wesley Professional, 7th edition, 2009.

[148] George Smith and David A. Atchison. *The Eye and Visual Optical Instruments*. Cambridge University Press, 1997. Cambridge Books Online.

[149] Hans Strasburger, Ingo Rentschler, and Martin Jüttner. Peripheral vision and pattern recognition: A review. *Journal of Vision*, 11(5):13, 2011.

[150] Adobe Systems. Adobe RGB (1998) Color Image Encoding. URL: `http://www.adobe.com/digitalimag/adobergb.html`, 1998.

[151] Susanne Tak. *Understanding and Supporting Window Switching*. PhD thesis, University of Canterbury, 2011.

[152] Leong-Hwee Teo, Bonnie John, and Marilyn Blackmon. CogTool-Explorer: A Model of Goal-directed User Exploration That Considers Information Layout. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2479–2488, New York, NY, USA, 2012. ACM.

[153] Martin Tomitsch, Karin Kappel, Andreas Lehner, and Thomas Grechenig. Towards a taxonomy for ambient information systems. pages 42–47, 2007.

[154] Anne M. Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97 – 136, 1980.

[155] Sally Turner, Julie Wylde, Martin Langham, and Andrew Morrow. Determining optimum flash patterns for emergency service vehicles: An experimental investigation using high definition film. *Applied ergonomics*, 45(5):1313–1319, 2014.

[156] Ronald van den Berg, Frans W Cornelissen, and Jos BTM Roerdink. A crowding model of visual clutter. *Journal of Vision*, 9(4):24, 2009.

[157] J. A. Vayghan, S. M. Garfinkle, C. Walenta, D. C. Healy, and Z. Valentin. The internal information transformation of IBM. *IBM Systems Journal*, 46(4):669–683, 2007.

[158] Bret Victor. A Brief Rant on the Future of Interaction Design. URL: `http://worrydream.com/ABriefRantOnTheFutureOfInteractionDesign/`, November 2011.

[159] Colin Ware. *Information Visualization – Perception for Design*. Morgan Kaufmann, 2000.

[160] Colin Ware, Joseph Bonner, William Knight, and Rod Cater. Moving Icons As A Human Interrupt. *International Journal of Human-Computer Interaction*, 4(4):341–348, 1992.

[161] Andrew B Watson et al. Temporal Sensitivity. *Handbook of Perception and Human Performance*, 1:6–1, 1986.

[162] David Whitney and Dennis M Levi. Visual Crowding: A fundamental limit on conscious perception and object recognition. *Trends in cognitive sciences*, 15(4):160–168, 2011.

[163] Christopher D. Wickens. *Engineering Psychology and Human Performance*. Harper Collens, New York, 2nd edition edition, 1992.

[164] Wikipedia. Electroencephalography. URL: `http://en.wikipedia.org/wiki/Electroencephalography`, 2015.

[165] Wikipedia. Lab Color Space. URL: `https://en.wikipedia.org/wiki/Lab_color_space`, 2016.

[166] Wikipedia. RGB Color Space. URL: `https://en.wikipedia.org/wiki/RGB_color_space`, 2016.

[167] Wikipedia. Ceiling effect (statistics). URL: `https://en.wikipedia.org/wiki/Ceiling_effect_(statistics)`, 2017.

[168] Wikipedia. Electrodermal Activity. URL: `https://en.wikipedia.org/wiki/Electrodermal_activity`, 2017.

[169] Wikipedia. List of common resolutions. URL: `https://en.wikipedia.org/wiki/List_of_common_resolutions`, 2017.

[170] Wikipedia. Sawtooth Wave. URL: `https://en.wikipedia.org/wiki/Sawtooth_wave`, March 2017.

[171] Wikipedia. Square Wave. URL: `https://en.wikipedia.org/wiki/Square_wave`, 2017.

[172] Wikipedia. Triangle Wave. URL: `https://en.wikipedia.org/wiki/Triangle_wave`, March 2017.

[173] Wesley Willett, Jeffrey Heer, and Maneesh Agrawala. Scented Widgets: Improving navigation cues with embedded visualizations. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1129–1136, 2007.

[174] Richard Williams. *The Animator's Survival Kit–Revised Edition: A Manual of Methods, Principles and Formulas for Classical, Computer, Games, Stop Motion and Internet Animators*. Faber & Faber, Inc., 2009.

[175] Changxu Wu and Yili Liu. Development and evaluation of an ergonomic software package for predicting multiple-task human performance and mental workload in human–machine interface design and evaluation. *Computers & Industrial Engineering*, 56(1):323–333, 2009.

[176] Steven Yantis and John Jonides. Abrupt visual onsets and selective attention: Voluntary versus automatic allocation. *Journal of Experimental Psychology: Human perception and performance*, 16(1):121, 1990.

[177] SN Yendrikhovskij, FJJ Blommaert, and H De Ridder. Optimizing color reproduction of natural images. In *Color and Imaging Conference*, volume 1998, pages 140–145. Society for Imaging Science and Technology, 1998.

Appendices

A Pilot Studies and Unsuccessful Methods for Measuring Task Performance in Dual-Task Experiments

We developed and tested many different tasks to find a suitable primary task to use in dual-task studies of the noticeability and distraction of highlighting techniques. Unlike prior dual-task studies where the primary tasks were mainly used to keep participants occupied (e.g. [28, 57, 99]), we wanted to use the primary task to collect *task performance* data (from which measures of distraction could be computed). To satisfy this requirement, there needed to be a frequent/continuous stream of measurable events. We hypothesized that distraction events may be short (i.e. sub-second) deviations in task performance.

A.1 Pilot Testing – Methods, Materials, and Set-Up

We conducted multiple rounds of pilot testing to verify, understand, and refine the experiment methods used in this thesis. For example, pilot studies were conducted to determine whether it was feasible to use a particular method (e.g. for the methods described in Section A.2), to fine-tune the number and difficulty of the tasks being performed by participants (e.g. for determining the disc radius used in Study 1, or the dot size/speed and peak-density for the dot-following steering task in Study 2), and to verify that all the experiment software and analysis routines worked correctly. Due to the exploratory nature of these tests and the need for multiple rounds of rapid iteration, pilot studies were conducted with a small group of 1-5 participants (namely, the author, his supervisor, and several other members of the HCI lab).

A.2 Examples of Unsuccessful Tasks

This section documents some of the tasks we tried, and the reasons why they were unsuccessful.

A.2.1 Ball following tasks (Random Walk)

The Task – Participants were asked to keep the cursor within a 20×20 px ball. The ball was located within a 200×200 pixel box in the center of the screen. Every 30 ms, the ball would move 5 px or less in a random direction within the box.

The Problem – Participant performance with this task is highly variable, even in the absence of highlights. Specifically, during a 10-30 second period, there would be statistically significant differences between error rates (i.e. distances from cursor to target) during one 5-15 period and another consecutive 5-15 period, implying that the technique would detect apparent "significant effects" when a highlighting technique was displayed, even if those effects were not caused by the highlighting technique. Furthermore, different participants either found the task easy or hard to perform. Also, the performance of a given participant would be significantly different at different times in the day.

Variations on Procedure – Several different variations/tweaks on the task setup were also tested. These included varying the size of the ball (e.g. 5px, 10px, and 30px versions were also tested), varying the speed of movement, and showing the ball with a line between the midpoint and cursor as a feedback mechanism for task performance.

A.2.2 Ball following tasks (Along Line)

The Task – Participants were again asked to track the movements of a ball that was travelling up and down in a straight line in the middle of the screen.

The Problem – Because the ball movements were predictable, participants could effectively go into "autopilot". As no conscious thought or mental effort was required to perform the task, participants could just continue to perform the movement task while mentally focussing on something else. Furthermore, the moving target position meant caused problems when measuring noticeability, as the cursor and/or the participants' visual focus would not have been in a fixed location relative to the target when the highlights first appeared. Also, some participants may have been inclined to keep following the line until the end of the current movement before changing course.

A.2.3 Single-digit sums

The Task – Participants were shown two single-digit numbers, and were asked to key in (using the numpad) the sum of those numbers. The answer would be a single digit answer.

The Problem – The ability to perform mental arithmetic and/or to recall specific results was a significant confound here. For instance some sums ("$x + 0$" or "$x + 1$") were generally easier to recall than others (e.g. "$2 + 5$" or "$7 + 2$"). As a result, response times for these tasks were highly variable, and occurred at a low/slow rate (i.e. averaging about one trial every 1-2 seconds).

A.2.4 Keying in matching letters/digits

The Task – Participants were shown a single letter or a digit in the middle of the screen, and were required to quickly press the key corresponding to that character. Participants would be continually shown a series of these characters until the end of the trial.

The Problem – While some conscious effort was needed to make quick decisions about what key to press, there was still the confound of familiarity with keyboard layout (i.e. participants could be tempted to look down at the keyboard). Also, we found that the latency between each keypress was sufficiently long that it was not clear whether any distraction effects may have occurred.

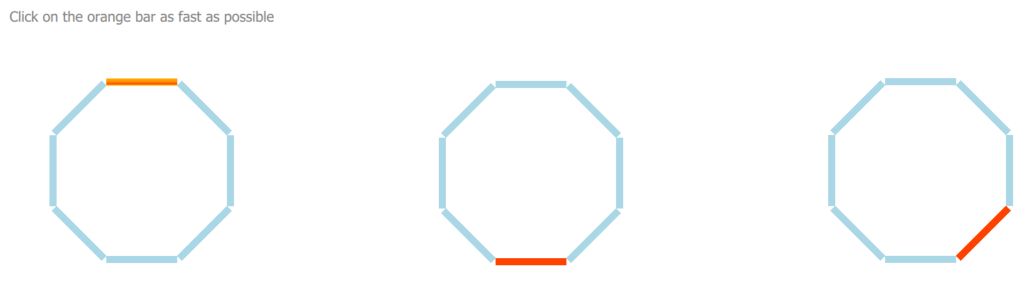
A.2.5 Matching Coloured Buttons

The Task – Participants would be shown a coloured square with three coloured buttons below it (labelled 1, 2, and 3). The task was to quickly click on whichever button matched the colour of the square above.

The Problem – Although this solved the problem of participants trying to look down at the keyboard to find the key to press, it ended up being harder for participants to perform. As a result, task performance was quite variable (e.g. having the same button twice in a row would be faster than having to move the mouse past two buttons to click on the right one).

A.2.6 Octagonal Fitts Task

The Task – Participants were presented with a set of 8 line segments arranged in an octagonal pattern (Figure A.1). All lines were the same length and thickness, and were equidistant from the center of the octagon. This was done to ensure that the *Index of Difficulty* [68] for each target was always constant. An octagonal pattern was used to ensure that there could be pairs of parallel lines to move between (even if this was not always done, to keep the task unpredictable).



(a) Step 0 – Start of Task (b) Step 1 – Directly Opposite (c) Step 1 – Opposite (R)

Figure A.1: *Screenshots of Octagonal Fitts Task*

Participants were instructed to "click on the orange bar". When the highlighted line was clicked, a line on the 'opposite' side of the octagon would get highlighted (e.g. if the top line was highlighted (Figure A.1(a)), the next target could be the bottom line (Figure A.1(b)), or the ones on either side of it (Figure A.1(c))). This process would repeat until the block/trial ended.

The activation areas for each target were larger than the visible bars (i.e. $1.25W$ and $20H$ respectively) so that participants did not need to waste time and effort trying to precisely target each line.

The Problem – As expected, participants were able to quickly perform this task at a steady pace. However, the click events were still too sparse to get the level of granularity desired for measuring subtle variations in task performance.

A.3 Task Design Insights

From these tasks, we realised that if the measureable events (e.g. keypresses, clicks, movement of the mouse, or some other task-specific metric) did not naturally occur frequently and regularly enough, there was a risk that the experiment is not sensitive enough to detect any subtle variations in human performance. This is because the effects of the distractors would likely be drowned out by noise arising from variations in inherent task difficulty and individual differences in problem solving/gameplay ability (e.g. some “game” tasks may be easier to perform than others, or some participants may particularly excel at certain types of tasks).

Although prior dual-task highlighting studies made extensive use of interactive games (e.g. tetris [28], card games [29], and driving simulators [57]) task performance in such games is highly variable, and present multiple hard-to-control challenges to the internal validity of highlighting experiments. As a result, the sensitivity of any empirical measures used would likely be further reduced if interactive games were used as the primary task. Bartram [28] cited similar concerns (e.g. “what is a ‘better’ Solitare strategy than another”) as one of the reasons why their dual-task study only used subjective measures.

B Study 1 – Consent Forms and Information Sheets

The following 3 pages show the *Information Sheet* and *Informed Consent Form* filled out by participants before taking part in the study, and the *flyer* used to recruit participants.

Total Pages: 3

HCI and Multimedia Lab
Department of Computer Science and Software Engineering
Telephone: +64 3364 2362 x4056
Email: joshua.leung@pg.canterbury.ac.nz, andy@cosc.canterbury.ac.nz

March 2016

Understanding Highlighting Techniques (HEC 2015/130)

Information Sheet for Participants

We (Joshua Leung and Prof Andy Cockburn) are carrying out an investigation into the effectiveness of highlighting techniques in user interfaces.

If you choose to take part in this study, your involvement in this project will be to carry out a series of short trials where you will use a computer mouse to drag a disc on to a target item as quickly as possible. There will be opportunities to take short breaks between trials; you will also be asked to take longer breaks at regular intervals during the experiment. Your hand and eye movements will be recorded to help us compare the highlighting techniques being investigated. At the conclusion of the experiment, you will be asked to fill out a short survey rating the highlighting techniques you have seen. In total, your participation should take approximately 20-30 minutes only.

Important: DO NOT proceed with this experiment if you are or have previously been susceptible to epileptic seizures. This experiment features rapidly flashing and flickering effects which may induce epileptic seizures in some people.

Participation is voluntary and you have the right to withdraw at any stage without penalty. **This is not in any way a test of your competence with computers.** You may ask for your raw data to be returned to you or destroyed at any point. If you withdraw, I will remove information relating to you. However, once analysis of raw data starts May 2016, it will become increasingly difficult to remove the influence of your data on the results.

The results of the project may be published, but you may be assured of the complete confidentiality of data gathered in this investigation: your identity will not be made public without your prior consent. To ensure anonymity and confidentiality, all references to participants in the investigation and data gathered will be anonymous. Only the signed consent forms will have information about your identity; these forms will be stored in a locked filing cabinet in Prof Cockburn's locked office for up to 10 years. All other raw data will be stored on password protected computer accounts accessible only to Joshua and Prof Cockburn. The raw data will be processed and used to compute metrics of highlighting effectiveness. The metrics computed may be included as part of a PhD thesis, publically available peer-reviewed research paper(s), and form the basis of a reference dataset of highlighting effects that may be distributed along with these documents as a valuable resource for guiding future research in the Human-Computer Interaction field. A thesis is a public document and will be available through the UC Library.

The project is being carried out as a requirement for a PhD in Computer Science by Joshua Leung under the supervision of Professor Andy Cockburn, who can be contacted at joshua.leung@pg.canterbury.ac.nz and andy@cosc.canterbury.ac.nz respectively. They will be pleased to discuss any concerns you may have about participation in the project.

This project has been reviewed and approved by the University of Canterbury Human Ethics Committee (application number: **2015/130**). Participants should address any complaints to The Chair, Human Ethics Committee, University of Canterbury, Private Bag 4800, Christchurch (human-ethics@canterbury.ac.nz).

If you agree to participate in the study, you are asked to complete the consent form on the next page. Please indicate to the researcher on the consent form if you would like to receive a copy of the summary of results of the project. Thank you for your co-operation.

HCI and Multimedia Lab
Department of Computer Science and Software Engineering
Telephone: +64 3364 2362 x4056
Email: joshua.leung@pg.canterbury.ac.nz, andy@cosc.canterbury.ac.nz

March 2016

Understanding Highlighting Techniques (HEC 2015/130)

Informed Consent Form for Participants

☐ I have been given a full explanation of this project and have had the opportunity to ask questions.

☐ I understand what is required of me if I agree to take part in the research.

☐ I understand that participation is voluntary and I may withdraw at any time without penalty. Withdrawal of participation will also include the withdrawal of any information I have provided should this remain practically achievable.

☐ I understand that this is not a test of my competence with computers in any way.

☐ I understand that any information or opinions I provide will be kept confidential to the researchers and that any published or reported results will not identify the participants. I understand that a thesis is a public document and will be available through the UC Library.

☐ I understand that all data collected for the study will be kept in locked and secure facilities and/or in password protected electronic form and will be destroyed after ten years.

☐ I understand the risks associated with taking part and how they will be managed. To the best of my knowledge, I am not aware that I have any susceptibility to epileptic seizures.

☐ I am not aware that I have red-green colour blindness.

☐ I understand that I am able to receive a report on the findings of the study by contacting the researcher at the conclusion of the project.

☐ I understand that I can contact the researcher Joshua Leung (joshua.leung@pg.canterbury.ac.nz) or supervisor Andy Cockburn (andy@cosc.canterbury.ac.nz) for further information. If I have any complaints, I can contact the Chair of the University of Canterbury Human Ethics Committee, Private Bag 4800, Christchurch (human-ethics@canterbury.ac.nz)

☐ I would like a summary of the results of the project.

☐ By signing below, I agree to participate in this research project.

Name: ________________________________ Signature: ___________________________

Date: ________________________________ Age: ______

Email Address (for report of findings, if applicable): ____

Highlighting Study: Survey
(Pid: ________)

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

Participants Needed for HCI Experiment

We are running an experiment to study the effectiveness of different highlighting techniques for use in user interfaces. Your task will be to perform a series of pointing tasks using a mouse.

The experiment takes 30 minutes to complete. You will be given a $10 voucher for Reboot café for your time.

If you are interested in participating, we will be running these experiments beginning April 2016 in Erskine Room 200 (head up the car park stairs to Level 2, enter the reception area, and find the first room on the left beside/behind the stairs). Head along to http://www.slyreply.com/app/sheets/o1ku41c5h5by/ to select a time and date which suits you.

Conditions:

- Participants **must be at least 18 years old (i.e. no STAR students)**
- Participants **should have normal or corrected vision**, and **should not have colour blindness (red-green)**.
- **DO NOT apply if you suffer from epilepsy or have had epileptic seizures. The effects used in these experiments feature multiple rapidly flashing and flickering graphics.**

If you have any questions, please contact us via the addresses below.

Thanks!

Joshua Leung
joshua.leung@pg.canterbury.ac.nz

Andy Cockburn
andy@cosc.canterbury.ac.nz

HCI and Multimedia Lab
Department of Computer Science and Software Engineering
University of Canterbury

C Study 2 – Consent Forms and Information Sheets

The following 3 pages show the *Information Sheet* and *Informed Consent Forms* filled out by participants at the start of the experiment. The *Information Sheet* was printed on a separate page, while the *Consent Form* (2 pages) was printed double-sided.

Total Pages: 3

HCI and Multimedia Lab
Department of Computer Science and Software Engineering
Telephone: +64 3364 2362 x4056
Email: joshua.leung@pg.canterbury.ac.nz, andy@cosc.canterbury.ac.nz

UC UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

January 2017

Understanding Highlighting Techniques (HEC 2015/130 – Amendment 1)

Information Sheet for Participants

We (Joshua Leung and Prof Andy Cockburn) are carrying out an investigation into the effectiveness of highlighting techniques in user interfaces.

If you choose to take part in this study, your involvement in this project will be to carry out a series of short trials where you will use a computer mouse to follow a moving dot while tapping the spacebar when you notice a highlighted window. There will be opportunities to take short breaks between trials. Your hand and eye movements will be recorded to help us compare the highlighting techniques being investigated. At the conclusion of the experiment, you will be asked to fill out a short survey rating the highlighting techniques you have seen. In total, your participation should take approximately 20-30 minutes only.

DO NOT proceed with this experiment if you are or have previously been susceptible to epileptic seizures. This experiment features rapidly moving/flashing effects which may induce epileptic seizures in some people.

Participation is voluntary and you have the right to withdraw at any stage without penalty. **This is not in any way a test of your competence with computers.** You may ask for your raw data to be returned to you or destroyed at any point. If you withdraw, I will remove information relating to you. However, once analysis of raw data starts March 2017, it will become increasingly difficult to remove the influence of your data on the results.

The results of the project may be published, but you may be assured of the complete confidentiality of data gathered in this investigation: your identity will not be made public without your prior consent. To ensure anonymity and confidentiality, all references to participants in the investigation and data gathered will be anonymous. Only the signed consent forms will have information about your identity; these forms will be stored in a locked filing cabinet in Prof Cockburn's locked office for up to 10 years. All other raw data will be stored on password protected computer accounts accessible only to Joshua and Prof Cockburn. The raw data will be processed and used to compute metrics of highlighting effectiveness. The metrics computed may be included as part of a PhD thesis, publically available peer-reviewed research paper(s), and form the basis of a reference dataset of highlighting effects that may be distributed along with these documents as a valuable resource for guiding future research in the Human-Computer Interaction field. A thesis is a public document and will be available through the UC Library.

The project is being carried out as a requirement for a PhD in Computer Science by Joshua Leung under the supervision of Professor Andy Cockburn, who can be contacted at joshua.leung@pg.canterbury.ac.nz and andy@cosc.canterbury.ac.nz respectively. They will be pleased to discuss any concerns you may have about participation in the project.

This project has been reviewed and approved by the University of Canterbury Human Ethics Committee (application number: **2015/130**). Participants should address any complaints to The Chair, Human Ethics Committee, University of Canterbury, Private Bag 4800, Christchurch (human-ethics@canterbury.ac.nz).

If you agree to participate in the study, you are asked to complete the consent form on the next page. Please indicate to the researcher on the consent form if you would like to receive a copy of the summary of results of the project. Thank you for your co-operation.

HCI and Multimedia Lab
Department of Computer Science and Software Engineering
Telephone: +64 3364 2362 x4056
Email: joshua.leung@pg.canterbury.ac.nz, andy@cosc.canterbury.ac.nz

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

January 2017

Understanding Highlighting Techniques (HEC 2015/130 – Amendment 1)
Informed Consent Form for Participants

☐ I have been given a full explanation of this project and have had the opportunity to ask questions.

☐ I understand what is required of me if I agree to take part in the research.

☐ I understand that participation is voluntary and I may withdraw at any time without penalty. Withdrawal of participation will also include the withdrawal of any information I have provided should this remain practically achievable.

☐ I understand that this is not a test of my competence with computers in any way.

☐ I understand that any information or opinions I provide will be kept confidential to the researchers and that any published or reported results will not identify the participants. I understand that a thesis is a public document and will be available through the UC Library.

☐ I understand that all data collected for the study will be kept in locked and secure facilities and/or in password protected electronic form and will be destroyed after ten years.

☐ I understand the risks associated with taking part and how they will be managed. To the best of my knowledge, I am not aware that I have any susceptibility to epileptic seizures.

☐ I am not aware that I have red-green colour blindness.

☐ I understand that I am able to receive a report on the findings of the study by contacting the researcher at the conclusion of the project.

☐ I understand that I can contact the researcher Joshua Leung (joshua.leung@pg.canterbury.ac.nz) or supervisor Andy Cockburn (andy@cosc.canterbury.ac.nz) for further information. If I have any complaints, I can contact the Chair of the University of Canterbury Human Ethics Committee, Private Bag 4800, Christchurch (human-ethics@canterbury.ac.nz)

☐ I would like a summary of the results of the project.

☐ By signing below, I agree to participate in this research project.

Name: ___________________________________ Signature: _____________________________

Date: __________________________________ Age: ______

Email Address (for report of findings, if applicable): ____

Highlighting Study: Survey
(Pid: ________)

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

1. How many windows (on average) do you typically have open on your computer?

2. How long do you spend playing games (e.g. mobile, console, PC, etc.) a week?

0-5 hours, 5-10 hours, 10-20 hours, 20+ hours.